



Etude de la synchronisation et de la stabilité d'un réseau d'oscillateurs non linéaires. Application à la conception d'un système d'horlogerie distribuée pour un System-on-Chip (projet HODISS).

Niamba Jean-Michel Akre

► **To cite this version:**

Niamba Jean-Michel Akre. Etude de la synchronisation et de la stabilité d'un réseau d'oscillateurs non linéaires. Application à la conception d'un système d'horlogerie distribuée pour un System-on-Chip (projet HODISS).. Autre. Supélec, 2013. Français. NNT : 2013SUPL0001 . tel-00789365

HAL Id: tel-00789365

<https://theses.hal.science/tel-00789365>

Submitted on 18 Feb 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N° d'ordre : 2013-01-TH

THÈSE DE DOCTORAT

DOMAINE : STIC
Spécialité : Électronique

**École Doctorale « Sciences et Technologies de l'Information des
Télécommunications et des Systèmes »**

Présentée par :

Jean-Michel N. AKRÉ

Sujet :

**Étude de la synchronisation et de la stabilité d'un réseau
d'oscillateurs non-linéaires. Application à la conception d'un
système d'horlogerie distribuée pour un System-on-Chip
(Projet HODISS).**

Soutenue le 11/01/2013 devant les membres du jury :

M. Pascal ACCO	Maître de Conférences, Université de Toulouse	Examineur
M. Éric COLINET	HDR, CEA-LETI, Minattec	Invité
M. Dominique DALLET	Professeur, Lab. IMS, IPB Bordeaux	Rapporteur
M. Dimitri GALAYKO	HDR, UPMC, Paris VI	Examineur
M. Jérôme JUILLARD	Professeur, École Supérieure d'Électricité	Directeur de Thèse
M. Abdel LISSER	Professeur, Université Paris-Sud XI	Examineur
M. Gérard SCORLETTI	Professeur, École Centrale de Lyon	Rapporteur



N° d'ordre : 2013-01-TH

THÈSE DE DOCTORAT

DOMAINE : STIC
Spécialité : Électronique

**École Doctorale « Sciences et Technologies de l'Information des
Télécommunications et des Systèmes »**

Présentée par :

Jean-Michel N. AKRÉ

Sujet :

**Étude de la synchronisation et de la stabilité d'un réseau
d'oscillateurs non-linéaires. Application à la conception d'un
système d'horlogerie distribuée pour un System-on-Chip
(Projet HODISS).**

Soutenue le 11/12/2013 devant les membres du jury :

M. Pascal ACCO	Maître de Conférences, Université de Toulouse	Examineur
M. Éric COLINET	HDR, CEA-LETI, Minattec	Invité
M. Dominique DALLET	Professeur, Lab. IMS, IPB Bordeaux	Rapporteur
M. Dimitri GALAYKO	Professeur assistant, UPMC, Paris VI	Examineur
M. Jérôme JUILLARD	Professeur, École Supérieure d'Électricité	Directeur de Thèse
M. Abdel LISSER	Professeur, Université Paris-Sud XI	Examineur
M. Gérard SCORLETTI	Professeur, École Centrale de Lyon	Rapporteur

A toute ma famille, à ma mère en particulier.

REMERCIEMENTS

Mes remerciements iront tout d'abord à mon directeur de thèse, Jérôme Juillard pour m'avoir donné la chance d'intégrer cette prestigieuse école qu'est Supélec. Sa relecture méticuleuse de chacun des chapitres m'ont permis de produire, je l'espère, un travail de bon aloi. Quoi qu'il en soit, j'ai beaucoup appris et je serai personnellement toujours fier de l'ensemble du travail qui a pu être réalisé.

Je remercie également le coordinateur du projet HODISS, Dimitri Galayko pour ses suggestions et conseils pertinents qui m'ont permis de préciser mon propos. Je tiens de même à le remercier pour son accueil toujours chaleureux, sa facilité de contact, d'échange, bref pour ses qualités humaines. Je voudrais par ailleurs lui exprimer ici toute ma gratitude pour l'hospitalité dont il a fait preuve envers moi durant mon séjour au sein de son équipe.

J'adresse un vif remerciement à Monsieur Eric Colinet pour sa présence au sein de mon comité de jury et m'enorgueillis de sa collaboration dans le projet HODISS, en particulier pour ses grandes qualités scientifiques et sa valeur.

Je voudrai aussi remercier profondément Messieurs Dominique Dallet et Gérard Scorletti qui m'ont fais l'honneur de rapporter cette thèse. Leurs remarques constructives sur le contenu de mon manuscrit m'ont permis d'y apporter des améliorations significatives.

Mes remerciements iront de même à Messieurs Pascal Acco et Abdel Lisser pour l'intérêt qu'ils ont porté à mon sujet de recherche, pour avoir accepté d'examiner mon travail et de participer à mon jury de soutenance.

Enfin j'aimerai dire un grand merci à mes parents et mes amis qui m'ont toujours épaulé et qui m'ont donné la force de terminer ce travail. Maman, Marie, je vous dois beaucoup.

RÉSUMÉ

Le projet HODISS dans le cadre duquel s'effectue nos travaux adresse la problématique de la synchronisation globale des systèmes complexes sur puce (System-on-Chip ou SOC, par exemple un multiprocesseur monolithique). Les approches classiques de distribution d'horloges étant devenues de plus en plus obsolètes à cause de l'augmentation de la fréquence d'horloge, l'accroissement des temps de propagation, l'accroissement de la complexité des circuits et les incertitudes de fabrication, les concepteurs s'intéressent (pour contourner ces difficultés) à d'autres techniques basées entre autres sur les oscillateurs distribués. La difficulté majeure de cette dernière approche réside dans la capacité d'assurer le synchronisme global du système. Nous proposons un système d'horlogerie distribuée basé sur un réseau d'oscillateurs couplés en phase. Pour synchroniser ces oscillateurs, chacun d'eux est en fait une boucle à verrouillage de phase qui permet ainsi d'assurer un couplage en phase avec les oscillateurs des zones voisines. Nous analysons la stabilité de l'état synchrone dans des réseaux cartésiens identiques de boucles à verrouillage de phase entièrement numériques (ADPLLs). Sous certaines conditions, on montre que l'ensemble du réseau peut synchroniser à la fois en phase et en fréquence. Un aspect majeur de cette étude réside dans le fait que, en l'absence d'une horloge de référence absolue, le filtre de boucle dans chaque ADPLL est piloté par les fronts montants irréguliers de l'oscillateur local et, par conséquent, n'est pas régi par les mêmes équations d'état selon que l'horloge locale est avancée ou retardée par rapport au signal considéré comme référence. Sous des hypothèses simples, ces réseaux d'ADPLLs dits "auto-échantillonnés" peuvent être décrits comme des systèmes linéaires par morceaux dont la stabilité est notoirement difficile à établir. L'une des principales contributions que nous présentons est la définition de règles de conception simples qui doivent être satisfaites sur les coefficients de chaque filtre de boucle afin d'obtenir une synchronisation dans un réseau cartésien de taille quelconque. Les simulations transitoires indiquent que cette condition nécessaire de synchronisation peut également être suffisante pour une classe particulière d'ADPLLs "auto-échantillonnés".

ABSTRACT

The HODISS project, context in which this work is achieved, addresses the problem of global synchronization of complex systems-on-chip (SOCs, such as a monolithic multiprocessor). Since the traditional approaches of clock distribution are less used due to the increase of the clock frequency, increased delay, increased circuit complexity and uncertainties of manufacture, designers are interested (to circumvent these difficulties) to other techniques based among others on distributed synchronous clocks. The main difficulty of this latter approach is the ability to ensure the overall system synchronization. We propose a clock distribution system based on a network of phase-coupled oscillators. To synchronize these oscillators, each is in fact a phase-locked loop which allows to ensure a phase coupling with the nearest neighboring oscillators. We analyze the stability of the synchronized state in Cartesian networks of identical all-digital phase-locked loops (ADPLLs). Under certain conditions, we show that the entire network may synchronize both in phase and frequency. A key aspect of this study lies in the fact that, in the absence of an absolute reference clock, the loop-filter in each ADPLL is operated on the irregular rising edges of the local oscillator and consequently, does not use the same operands depending on whether the local clock is leading or lagging with respect to the signal considered as reference. Under simple assumptions, these networks of so-called “self-sampled” all-digital phase-locked-loops (SS-ADPLLs) can be described as piecewise-linear systems, the stability of which is notoriously difficult to establish. One of the main contributions presented here is the definition of simple design rules that must be satisfied by the coefficients of each loop-filter in order to achieve synchronization in a Cartesian network of arbitrary size. Transient simulations indicate that this necessary synchronization condition may also be sufficient for a specific class of SS-ADPLLs.

GLOSSAIRE

ADPLL: All-Digital Phase-Locked Loop

ANR-ARFU: Programme Architectures du Futur de l'Agence Nationale de Recherche

ASIC: Application-specific integrated circuit

BB-DPD: Bang-Bang Digital Phase Detector

CEA : Commissariat à l'énergie atomique et aux énergies alternatives

CMOS: Complementary Metal Oxide Semiconductor

CQLF: Common Quadratic Lyapunov Function

DC: Direct current Level

DCO: Digitally Controlled Oscillator

DPD: Digital Phase Detector

FPGA: Field-Programmable Gate Array

HODISS: Horlogerie Distribuée pour les SoCs Synchrones

LETI: Laboratoire d'Electronique des Technologies de l'Information

LIP6: Laboratoire d'Informatique de Paris 6

LMI: Linear Matrix Inequality

LSI: Large Scale Integration

MATLAB: Matrix Laboratory

ODE: Ordinary Differential Equations

PD: Phase Detector

PI (Filtre): Filtre Proportional Integral

PLL: Phase-Locked Loop

PLS: Piecewise Linear System

PQLF: Piecewise Quadratic Lyapunov Function

SoC: System on Chip

SS-ADPLL: Self-Sampled All-Digital Phase-Locked Loop

SSE: Signaux et Systèmes Electroniques

SUPÉLEC: École Supérieure d'Électricité

TDC: Time to Digital Converter

VCO: Voltage Controlled Oscillator

VLSI: Very-large-scale integration

XOR PLL: Phase-Locked Loop with a XOR gate as detector

TABLE DES MATIÈRES

INTRODUCTION GÉNÉRALE	1
0.1. Contexte	2
0.2. Le projet HODISS	4
0.3. Objectif et organisation de la thèse	5
PARTIE I : LA BOUCLE A VERROUILLAGE DE PHASE	8
Chapitre 1 :	
La boucle à verrouillage de phase analogique	9
1.1. Introduction.....	9
1.2. Définitions.....	9
1.3. Modélisation du comparateur	10
1.4. Filtre passe bas et VCO	11
1.5. Équation différentielle du système.....	12
1.6. Dynamique normalisée	14
1.7. Étude des trajectoires dans le plan de phase et paramétrage du système	16
1.7.1. Positions d'équilibre	17
1.7.2. Analyse des portraits de phase.....	17
1.8. Conclusion.....	19
Chapitre 2 :	
La boucle à verrouillage de phase digitale	21
2.1. Introduction.....	21
2.2. Classification des ADPLLs	22
2.2.1. L'ADPLL avec un comparateur XOR.....	23
2.2.2. Les ADPLLs avec comparateurs à bascules.....	26
2.2.2.1. Principe.....	26
2.2.2.2. Exemple de conception d'une ADPLL, modèle Simulink.....	30
2.3. L'ADPLL auto-échantillonnée (SS-ADPLL)	31
2.3.1. Le DPD mis en œuvre dans le projet HODISS.....	32
2.3.1.1. Le détecteur bang-bang (BB-DPD).....	32
2.3.1.2. Fonctionnement du TDC	34
2.3.2. Le filtre proportionnel-intégral (PI).....	36
2.3.3. L'oscillateur à commande digitale (DCO)	37
2.3.4. Description d'un nœud sans auto-échantillonnage	40
2.3.5. Modélisation de la SS-ADPLL et effet self-sampling.....	42
2.3.5.1. Les SS-ADPLLs passéistes.....	44
2.3.5.1.1. Les SS-ADPLLs passéistes de type A	46
2.3.5.1.2. Les SS-ADPLLs passéistes de type B	51
2.3.5.2. La SS-ADPLL nihiliste.....	54
2.4. Conclusion.....	56
Chapitre 3 :	
Étude de la stabilité au sens de Lyapunov	57
3.1. Introduction.....	57
3.2. Stabilité des systèmes discrets linéaires par morceaux.....	58
3.2.1. Stabilité quadratique	59
3.2.2. Stabilité quadratique par morceaux.....	60
3.2.3. Application aux SS-ADPLLs.....	62

3.2.3.1.	Stabilité d'un nœud de SS-ADPLL passéiste de type A.....	62
3.2.3.2.	Stabilité d'un nœud de SS-ADPLL passéiste de type B.....	63
3.3.	Conclusion.....	66
PARTIE II : LES RÉSEAUX DE BOUCLES A VERROUILLAGE DE PHASE DIGITALES.....		68
Chapitre 4 :		
Synchronisation des réseaux cartésiens de SS-ADPLLs.....		70
4.1.	Introduction.....	70
4.2.	Stabilité des réseaux cartésiens autonomes de SS-ADPLLs.....	70
4.2.1.	Réseaux de 2 SS-ADPLLs.....	70
4.2.1.1.	Réseaux de 2 SS-ADPLLs passéistes de type A.....	71
4.2.1.2.	Réseaux de 2 SS-ADPLLs passéistes de type B.....	72
4.2.1.3.	Réseaux de 2 SS-ADPLLs nihilistes de type A.....	72
4.2.1.4.	Réseaux de 2 SS-ADPLLs nihilistes de type B.....	73
4.2.2.	Réseaux passéistes de tailles quelconques	76
4.2.2.1.	Définition du cadre.....	76
4.2.2.2.	Discussion.....	84
4.2.3.	Illustrations et résultats	86
4.2.3.1.	Équations maîtresses des SS-ADPLLs passéistes de type A et B.....	86
4.2.3.2.	Équations maîtresses générales	87
4.2.3.3.	Domaines de stabilité de réseaux de SS-ADPLLs passéistes de type A et B.....	87
4.2.3.4.	Comportements en régimes transitoires de réseaux de SS-ADPLLs passéistes de type A et B	88
4.2.4.	Réseaux nihilistes de tailles quelconques	91
4.2.4.1.	Équations maîtresses des SS-ADPLLs nihilistes de type A et B.....	91
4.2.4.2.	Domaines de stabilité des réseaux de SS-ADPLLs nihilistes de type A et B.....	92
4.3.	Les systèmes moyens comme outils d'optimisation.....	94
4.4.	Sensibilité au bruit de quantification	97
4.5.	Réseaux cartésiens avec référence	100
4.6.	Conclusion.....	101
Chapitre 5 :		
Validation des résultats théoriques.....		102
5.1.	Introduction.....	102
5.2.	Simulation de réseaux passéistes en présence de non-idéalités	102
5.2.1.	Réseaux passéistes de type B	104
5.3.	Prototype FPGA d'un réseau de SS-ADPLLs.....	109
5.3.1.	Présentation de l'architecture du système testé.....	110
5.3.2.	Mesures expérimentales	111
5.4.	Conclusion.....	113
CONCLUSIONS GÉNÉRALES ET PERSPECTIVES		115
ANNEXES		118
A.1.	Les cycles limites pour une PLL analogique non linéaire.....	118
A.2.	Stabilité des réseaux autonomes d'ADPLLs idéaux.....	120
A.2.1.	Réseaux complètement connectés	122
A.2.2.	Réseaux cartésiens	123
A.3.	Simulation de réseaux de SS-ADPLLs de type A en présence de non-idéalités	126
A.4.	Génération des matrices d'adjacence.....	128
A.5.	Simulation d'un réseau de SS-ADPLLs.....	131

A.6. Simulation de réseaux de SS-ADPLLs (prise en compte de la quantification et de la saturation).....	134
A.7. Détermination de Fonctions de Lyapunov pour des systèmes linéaires par morceaux.....	138
LISTE DES PUBLICATIONS.....	140
Article de revue avec comité de lecture.....	140
Conférences Internationales avec Actes et Comité de Lecture.....	140
RÉFÉRENCES BIBLIOGRAPHIQUES	141
LISTE DES FIGURES ET TABLEAUX	150
RÉSUMÉ.....	154
ABSTRACT	154

INTRODUCTION GÉNÉRALE

0.1. Contexte

Distribuer une horloge à travers les extensions physiques d'un système synchrone numérique de telle sorte que la référence de temps pour toutes les communications internes et les fréquences de toutes les horloges distribuées soient rigoureusement les mêmes (alignement des horloges) est une contrainte difficile à assurer. Traditionnellement, cette tâche est accomplie par la distribution de la sortie d'un oscillateur central via un réseau en forme d'arbre ou grille, avec des répéteurs¹ à des intervalles nécessaires [1,3]. Bien que simple dans son concept, cette méthode souffre entre autre d'une mauvaise fiabilité et d'une forte gigue temporelle. En effet, la conception d'un système de distribution de type arbre impose de sévères contraintes de fabrication dans la mesure où un changement de la position des récepteurs ou des dimensions peut provoquer un déséquilibre de l'arbre entier. Plusieurs stratégies visant à réduire les effets non désirées de l'inexactitude d'une horloge par compensation de la gigue temporelle (skew) [107] ont été proposées dans la littérature [6-7] mais peu d'entre elles ont suscité un intérêt suffisamment important pour une mise en œuvre pratique.

Dans les systèmes sur puces (SoCs) synchrones de grandes tailles, les systèmes de distribution d'horloge basés sur des réseaux composés d'oscillateurs synchronisés [1,3,8] représentent une approche alternative aux méthodes classiques de distribution d'horloge de type arbre ou grille [2,8,10]. Cette approche alternative, présentée en 1995 par Pratt et Nguyen évite le routage de l'horloge à travers de longues lignes d'interconnexion et maintient la simplicité de fonctionnement synchrone sans souffrir des inconvénients de la synchronisation centralisée. Dans de tels systèmes, un réseau d'oscillateurs couplés prend la place de la source d'horloge centralisée, fournissant des signaux d'horloge séparés mais synchronisés pour des parties physiquement distantes du système. Il s'agit d'une architecture de distribution active permettant d'ajuster en temps réel, les différences des délais d'une horloge à l'autre [13].

L'action qui permet à deux ou plusieurs événements de se réaliser simultanément avec une période identique, autrement dit, le sujet de la *synchronisation* [97, 104] et, plus

¹ Dispositifs électroniques constitués de récepteurs et d'émetteurs, utilisés pour amplifier le signal et le faire repartir sans modifier son contenu. La présence de répéteurs dans un circuit électronique permet de compenser les pertes de transmission d'une ligne.

largement, le sujet du consensus multi-agent¹ a reçu un intérêt considérable durant ces dernières années. En 2007, Olfati-Saber, Fax et Murray ont fourni un cadre théorique pour l'analyse des algorithmes de consensus pour les systèmes multi-agents en réseau et ont discuté entre autre des liens avec les problèmes de consensus au sein des réseaux de systèmes dynamiques et des applications diverses incluant la synchronisation d'oscillateurs couplés [4]. De bons points d'initiation sur le sujet du consensus multi-agent peuvent être trouvés dans [4-5], ceux sur le sujet de la synchronisation dans [14], principalement dédié à de nombreux phénomènes de synchronisation observés dans la nature, et dans [15-16, 86], plus spécifiquement consacré à la synchronisation dans des réseaux.

En pratique, pour effectuer la synchronisation des oscillateurs, un processus dynamique de correction d'erreurs de type PLL (boucle à verrouillage de phase) garantit l'alignement de la phase globale en utilisant les comparaisons locales de phase entre un oscillateur quelconque et l'oscillateur local voisin de sorte à traiter efficacement le problème de la distribution de la référence de temps et de la fréquence sur une large surface.

Toutefois, en raison du fait que la phase d'un oscillateur est une grandeur modulaire, une mise en œuvre naïve des horloges distribuées synchrones peut souffrir de "mode-lock", c'est à dire le verrouillage de la phase d'un oscillateur local dans un état d'équilibre stable mais indésirable, où tous les oscillateurs ne sont pas en phase [1].

Plusieurs travaux se sont essayés à la résolution de ce problème important dont ceux de Pratt et Nguyen [1] en vue de permettre au concepteur de garantir que les échelles de temps et de fréquence de toutes les horloges soient alignées après un temps fini. Ils préconisent l'utilisation de comparateurs de phase particuliers dont la caractéristique présente une pente négative au-delà d'une différence de phase de $\pi/2$. La fiabilité de cette approche ayant été démontrée par Pratt et Nguyen, une mise en œuvre utilisant 16 PLLs, donnée en Fig.0.1 a été proposée par Gutnik et Chandrakasan en 2000 [3]. Leur contribution majeure a été de réaliser des comparateurs de phase approchant la caractéristique indiquée dans [1], c'est-à-dire dont le gain est négatif lorsque l'erreur de phase atteint un certain seuil (inférieur à $\pi/2$), éliminant ainsi les modes stables indésirables. Le comparateur de phase en question fut réalisé à partir d'un arbitre

¹ Compte tenu des valeurs initiales (scalaires ou vectorielles) des agents, parvenir à un consensus consiste en l'établissement de conditions pour lesquelles, grâce à des interactions locales, les agents arrivent asymptotiquement à un accord sur une valeur commune.

numérique et de deux générateurs d'impulsion. La sortie du comparateur est un courant impulsionnel dont la valeur moyenne alimente un filtre analogique. Néanmoins, la PLL ainsi mise en œuvre n'a pas échappé aux inconvénients associés à l'utilisation de techniques analogiques. Cela est certainement la raison pour laquelle notre étude bibliographique n'a pas révélé une importante utilisation de ce circuit par les concepteurs de circuits numériques, l'approche n'étant pas compatible avec les techniques conventionnelles de traitement de flux numériques. En 2003, O'Mahony et al. ont proposé une autre étude pouvant se classer dans la même catégorie que l'étude précédente du fait que l'électronique utilisée était elle aussi analogique [32]. Cette étude traite d'un réseau d'oscillateurs couplés fonctionnant comme un milieu continu dans lequel se propagent des ondes stationnaires. Un autre inconvénient de cette architecture réside dans le fait que l'amplitude d'oscillation diffère d'un nœud à l'autre, c'est-à-dire qu'elle est plus grande au milieu de la puce et plus petite près des bords. Cette dernière considération suggère la nécessité de mettre en place des dispositifs spéciaux pour le redressement de l'amplitude.

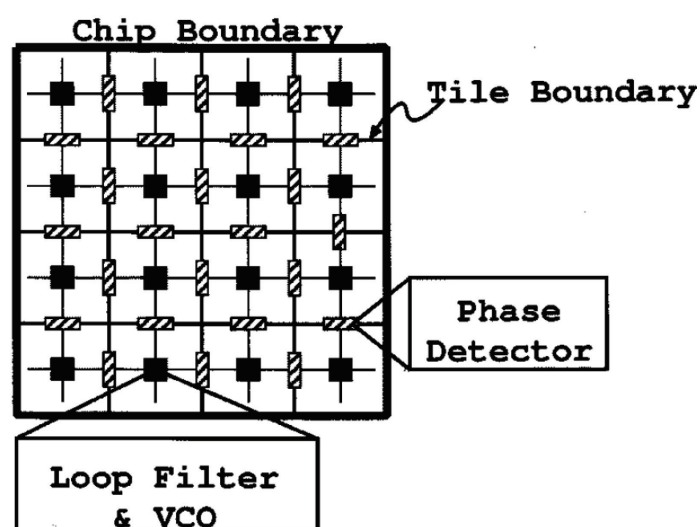


Fig. 0.1 – réseau de distribution d'horloge de 16 nœuds, réalisé par Gutnik et Chandrakasan.

0.2. Le projet HODISS¹

Le projet HODISS, financé par le programme “Architectures du Futur” (ARFU) de l'Agence Nationale de Recherche Française (ANR), adresse la problématique de la

¹ Horlogerie distribuée pour les SoCs synchrones.

synchronisation globale des systèmes complexes sur puce (System-on-Chip ou SoC, par exemple un multiprocesseur monolithique). Les incertitudes de fabrication, l'accroissement des temps de propagation, l'augmentation de la fréquence d'horloge et l'accroissement de la complexité des circuits issus des technologies VLSI modernes font que les concepteurs s'orientent vers des circuits constitués de multiples zones isochrones¹. C'est dans cet axe de recherche que le projet HODISS se situe et vise à poursuivre le travail présenté dans [1] et [3] afin de résoudre les difficultés majeures de cette approche qui sont les limitations induites par l'utilisation d'une électronique analogique. Il a pour but d'étudier et de concevoir un système de distribution d'horloge pour circuit numérique tout-synchrone basé sur un flux de conception entièrement numérique, afin d'être facilement intégrable, compatible avec les fonctions numériques des blocs du SoC et de bénéficier de l'immunité face au bruit dont bénéficient les composants numériques. Dans cette horlogerie distribuée, le couplage en phase de chaque oscillateur avec les zones isochrones voisines est assuré par des PLLs (Phase-Locked Loops), permettant ainsi la synchronisation de l'ensemble des oscillateurs. Ce projet a pour ambition un choix de l'architecture, une élaboration de son modèle mathématique, une analyse qualitative et quantitative de son comportement, une analyse de condition de stabilité [88] du fonctionnement synchrone et une conception du schéma électrique (netlist) du circuit prototype dans une technologie CMOS [87] avancée. Le projet vise aussi une réduction de la surface occupée par le système tout en utilisant une technique permettant de réduire la puissance dissipée par l'horlogerie du circuit [9]. Les partenaires du projet sont le LIP6², le département SSE (Signaux et Systèmes Électroniques) de Supélec³ et le DCIS (Département Conception et Intégration dans les Systèmes) du CEA-LETI⁴.

0.3. Objectif et organisation de la thèse

Il s'agit d'effectuer une étude théorique approfondie du système d'horlogerie constitué par un réseau de PLL couplées. Cette thèse est organisée de manière à pouvoir, en allant des notions les plus élémentaires aux plus complexes, mener une étude

¹ Surfaces accessibles à intervalles de temps égaux par un oscillateur unique.

² Laboratoire d'Informatique de l'université Paris 6.

³ École Supérieure d'Électricité, Gif-sur-Yvette.

⁴ Laboratoire d'électronique et des technologies de l'information du CEA, Grenoble.

complète autour des boucles à verrouillage de phase développées dans le cadre du projet HODISS. Cette étude comprend un travail théorique pour un nœud de PLL et pour des réseaux conçus en utilisant cette PLL comme bloc de base. Enfin elle comprend une partie implémentation permettant de valider l'approche théorique et de donner une connotation pratique à l'ensemble du travail réalisé. La thèse est organisée en deux parties, la première est consacrée à l'étude d'un nœud de PLL. Dans le chapitre premier de cette partie, nous nous intéressons d'abord à la boucle à verrouillage de phase analogique en vue d'étudier sa théorie et de clarifier toutes les zones d'ombres dues à son fonctionnement complexe et non-linéaire. La description des composants de boucle est donnée (détecteur de phase et de fréquence, oscillateur commandé et filtre). Même si l'étude réalisée dans cette partie porte sur le comportement d'une boucle d'ordre deux très largement utilisée, cette étude, qui peut être généralisée par la suite, fournit une interprétation physique du mécanisme de PLL et donne un aperçu du comportement non linéaire des boucles d'ordre supérieur. Les notions de base permettant d'expliquer le comportement dynamique d'une PLL (synchronisation, cycles limites, acquisition, zones de capture...) y sont exposées. Il est montré que la synchronisation est assurée par une action de contrôle (régulation qui se fait au niveau du filtrage) et un autre paramètre appelé *plage de verrouillage* qui représente la plage de fréquences à l'intérieur de laquelle on peut faire varier la fréquence du signal d'entrée de la boucle tout en préservant son aptitude à pouvoir verrouiller. Ainsi, des critères pour une conception optimum sont proposés.

Par la suite, dans le chapitre second traitant des boucles à verrouillage de phase digitales (ADPLLs), nous donnons une brève description des différents types d'ADPLLs. Les ADPLLs [89], considérées comme plus performantes que les PLLs analogiques ont un comparateur de phase de type digital. Dans les systèmes ADPLLs que nous allons traiter, un régulateur proportionnel intégral (PI) est placé après un comparateur de phase et de fréquence digital (DPD). Enfin nous expliquons comment l'on en est arrivé au choix de PLL étudié dans notre projet à savoir la boucle à verrouillage de phase digitale auto-échantillonnée (SS-ADPLL). Le terme « auto-échantillonnée » provient du fait que le filtre numérique est piloté par les fronts montants de l'horloge locale de la PLL. Le modèle est décrit en détail, tous les blocs composant le système sont modélisés et les équations d'états sont données. Le comparateur considéré ici est quasi linéaire pour de petites erreurs de phases et sature lorsque les erreurs de phase sont suffisamment

importantes. L'étude théorique est principalement développée en considérant la zone linéaire, car nous ne considérons que de petites variations autour des points d'équilibre stable. Ces considérations permettent d'établir une théorie en ce qui concerne ces SS-ADPLLs qui peuvent être finalement considérées comme étant des systèmes linéaires par morceaux (PLSs). Le troisième chapitre traite de la stabilité des SS-ADPLLs et permet d'obtenir des outils nécessaires pour l'analyse de ces PLSs. Il est aussi montré qu'un retard supplémentaire dans la boucle de retour provoque une réduction significative du domaine de stabilité.

La deuxième partie comprend deux chapitres et traite respectivement des réseaux de SS-ADPLLs et des vérifications expérimentales effectuées sur un réseau de 16 nœuds. Au quatrième chapitre, nous traitons deux types de réseaux (réseaux autonomes et réseaux avec une référence) selon deux types de configuration (réseaux complètement connectés et réseaux cartésiens). Les réseaux complètement connectés sont indiqués du fait de leur intérêt théorique et des travaux effectués sur ce sujet [33-34]. Cependant, l'implémentation d'un réseau complètement connecté de grande taille est peu réaliste. Les conditions de synchronisation de ces réseaux obtenues par des simulations transitoires sont néanmoins données. Une attention particulière est accordée par contre aux réseaux cartésiens, réseaux que nous avons choisi d'adopter eu-égard à des considérations pratiques liées à notre application. Une analyse des conditions de synchronisation de ces réseaux est investie. Par la suite, des simulations transitoires prenant en compte la non-linéarité du comparateur (saturations) permettent d'avoir des résultats plus proches du comportement réel du système. Enfin, au cinquième et dernier chapitre, un ensemble de vérifications expérimentales est réalisé afin de vérifier les approches théoriques développées dans le cadre du projet. Les tests sur un prototype FPGA de 16 nœuds permettent en définissant un certain nombre de spécifications, de rendre compte des performances observées. Les codes, les calculs mathématiques fastidieux et le rappel des principaux résultats sur lesquels s'appuient les théories développées dans ce mémoire sont présentés en annexe.

PARTIE I : LA BOUCLE A VERROUILLAGE DE PHASE

Les boucles à verrouillage de phase (PLL) ont été conçues en 1932 par Henri de Bellescize. Le principe fondamental de la PLL était en soi extrêmement intéressant mais la technologie de l'époque ne permettait pas de réaliser des montages simples et économiques. Il a fallu comme pour beaucoup de principes découverts et mis en attente, l'avènement des circuits de grande intégration LSI (Large Scale Integration). Depuis, ce circuit électronique est couramment utilisé dans des applications impliquant le contrôle automatique de la phase ou de la fréquence, telles que des applications en communication, en contrôle des systèmes, ou en mesures et instrumentations. Le principe consiste en l'asservissement d'un oscillateur de fréquence variable sur une source de référence pouvant provenir d'un quartz ou encore d'un signal modulé. Dans le cas d'une PLL analogique, l'oscillateur à asservir est commandé en tension (VCO : Voltage Controlled Oscillator) et dans le cas d'une PLL digitale, l'oscillateur est commandé par un mot numérique (DCO : Digitally Controlled Oscillator). Par ailleurs, le mécanisme de verrouillage de la phase est rencontré dans plusieurs oscillateurs biologiques comme par exemple le cœur ou les poumons [35].

Chapitre 1 :

La boucle à verrouillage de phase analogique

1.1. Introduction

Ce chapitre est consacré uniquement à l'étude de la PLL analogique. Le fait d'être amené à travailler avec des boucles à verrouillage de phase requiert inévitablement de comprendre le fonctionnement d'une PLL analogique classique, qui a fait l'objet d'un intérêt considérable soldé par une abondante littérature [1-3,32-35]. Comprendre ce système permet d'avoir la base intuitive et naturelle pour la compréhension de tous les autres types de PLL et permet à l'aide de la détermination des équations d'état du système, de décrire les principales propriétés des boucles à verrouillage de phase. Ce chapitre permettra de faire, à partir de l'analyse continue, un transfert en temps discret en vue de comprendre le fonctionnement élémentaire d'une PLL digitale. Disposant de tous ces éléments, on introduira la SS-ADPLL qui représente le nœud élémentaire de notre réseau (de SS-ADPLLs).

1.2. Définitions

Des abus de langage étant souvent observés dans le vocabulaire des signaux sinusoïdaux, nous allons ici essayer de définir avec précision les termes permettant de caractériser un signal sinusoïdal.

Soit $S(t)$ un signal sinusoïdal d'amplitude constante M :

$$S(t) = M \cdot \sin[\varphi(t)], \text{ avec } \varphi(t) = \omega_0 \cdot t + u(t) \quad (1.1)$$

La quantité $\varphi(t)$ est appelée **phase instantanée** du signal $S(t)$ et s'exprime en radians. La dérivée de la phase instantanée permet d'obtenir la **pulsation instantanée** de $S(t)$ tandis que la constante ω_0 est sa **pulsation absolue**. Dans le cas particulier où

l'écart $u(t)$ est constant, cet écart est appelé **phase absolue** et la pulsation instantanée se confond avec la pulsation absolue:

$$\frac{d\varphi(t)}{dt} = \omega_0 \quad (1.2)$$

Lorsque les termes de phase ou de pulsation sont employés sans plus de précision, il s'agit en fait de la phase ou de la pulsation absolue.

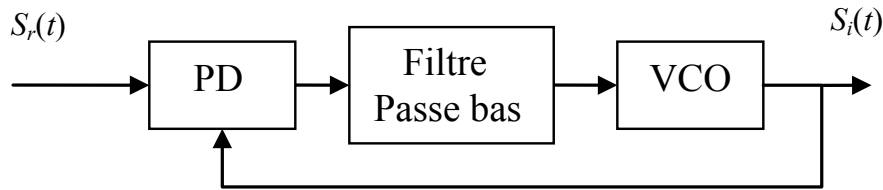


Fig. 1.1 – Schéma classique d'une PLL analogique.

Le schéma fonctionnel d'une PLL analogique est donné à la Fig.1.1, c'est un système asservi à retour unitaire avec en son entrée le signal sinusoïdal $S_r(t)$ et à sa sortie le signal sinusoïdal $S_i(t)$ définis comme suit :

$$S_r(t) = M_r \cdot \sin[\varphi_r(t)], \text{ avec } \varphi_r(t) = \omega_r \cdot t + u_r(t) \quad (1.3)$$

$$S_i(t) = M_i \cdot \sin[\varphi_i(t)], \text{ avec } \varphi_i(t) = \omega_i \cdot t + u_i(t) \quad (1.4)$$

Si l'on suppose que le signal de référence est de fréquence pure et très peu bruité (cas où l'on choisit un oscillateur de type quartz ou rubidium), alors on peut considérer que $u_r(t)$ est une constante que l'on notera U_r :

$$\varphi_r(t) = \omega_r \cdot t + U_r, U_r = cte \quad (1.5)$$

1.3. Modélisation du comparateur

Le comparateur a pour rôle d'extraire la différence de phase des signaux qu'on lui injecte en son entrée. Dans le cas d'une PLL analogique, cette tâche est souvent gérée par un multiplicateur. Le comparateur effectue une multiplication entre S_r et S_i et génère une erreur $\bar{\varepsilon}(t)$ proportionnelle à leur écart de phase.

$$\bar{\varepsilon}(t) = S_r(t) \cdot S_i(t) \quad (1.6)$$

En utilisant (1.3) et (1.4),

$$\bar{\varepsilon}(t) = M_r M_i \sin[\omega_r t + U_r] \cdot \sin[\omega_i t + u_i(t)]$$

qui se ramène à:

$$\bar{\varepsilon}(t) = \frac{1}{2} M_r M_i (\cos[(\omega_r - \omega_i)t + U_r - u_i(t)] - \cos[(\omega_r + \omega_i)t + U_r + u_i(t)]) \quad (1.7)$$

en utilisant la formule trigonométrique de développement.

On constate que la sortie du comparateur se présente sous la forme de deux composantes. Une composante DFC (*difference frequency component* $\omega_r - \omega_i$) et une composante haute fréquence HF $\omega_r + \omega_i$ appelée *ripple*.

1.4. Filtre passe bas et VCO

Le filtre est choisit de sorte à pouvoir entre autre éliminer la composante haute fréquence ($\omega_r + \omega_i$) et récupérer la composante DFC, qui sera appliquée à l'entrée de l'oscillateur contrôlé en tension (VCO). En nous mettant dans le cas de figure ou cette exigence est respectée, on peut dès lors utiliser comme erreur à la sortie du détecteur de phase, l'expression ε :

$$\begin{aligned} \varepsilon(t) &= \frac{1}{2} M_r M_i \cos[(\omega_r - \omega_i)t + U_r - u_i(t)] \\ \varepsilon(t) &= \frac{1}{2} M_r M_i \cos(\varphi_r - \varphi_i)(t) \end{aligned} \quad (1.8)$$

Le filtre que nous allons considérer dans cette étude est un filtre passe bas passif de premier ordre, avec un pôle et un zéro, donné à la Fig. 1.2.

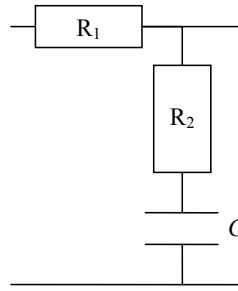


Fig. 1.2 – Filtre de boucle.

En appliquant le diviseur de tension, on obtient la fonction de transfert de ce filtre qui s'écrit :

$$H(s) = \frac{R_2 Cs + 1}{(R_1 + R_2)Cs + 1} \quad (1.9)$$

La relation (1.9) permet d'obtenir une première équation différentielle mettant en relation l'entrée du filtre à savoir l'erreur de phase $\varepsilon(t)$ et sa sortie $y(t)$ qui représente la commande du VCO :

$$y(t) + (R_1 + R_2)C \frac{dy(t)}{dt} = \varepsilon(t) + R_2 C \frac{d\varepsilon(t)}{dt} \quad (1.10)$$

Par définition, le VCO délivre un signal dont la pulsation instantanée est une fonction linéaire de la commande qu'on lui injecte.

$$\frac{d\varphi_i(t)}{dt} = \omega_i + g \cdot y(t) \quad (1.11)$$

La relation précédente signifie qu'un VCO peut être modélisé comme un intégrateur, la constante g représente son gain en fréquence.

1.5. Équation différentielle du système

Disposant maintenant de toutes les équations de transfert aux différents blocs de la PLL, nous allons pouvoir en les combinant, obtenir l'équation différentielle régissant le fonctionnement de l'ensemble du système. Les équations (1.3), (1.4) et (1.11) permettent d'écrire :

$$\frac{d\varphi_r(t)}{dt} - \frac{d\varphi_i(t)}{dt} = \omega_r - \omega_i - g \cdot y(t)$$

Si on note la différence de phase :

$$r(t) = \varphi_r(t) - \varphi_i(t) \quad (1.12)$$

Alors en utilisant (1.10), (1.11) et (1.12), nous obtenons le système :

$$\begin{cases} \frac{dr(t)}{dt} = -g \cdot y(t) + \omega_r - \omega_i \\ \frac{dy(t)}{dt} = \frac{1}{(R_1 + R_2)C} \left[\varepsilon(t) - y(t) + R_2 C \frac{d\varepsilon(t)}{dt} \right] \end{cases} \quad (1.13)$$

qui en posant $\begin{cases} K = \frac{1}{2} M_r M_i \\ \omega_{c1} = \frac{1}{(R_1 + R_2)C} \\ \omega_{c2} = \frac{1}{R_2 C} \end{cases}$

peut se réécrire :

$$\begin{cases} \frac{dr(t)}{dt} = -g \cdot y(t) + \omega_r - \omega_i \\ \frac{dy(t)}{dt} = \omega_{c1} \left\{ K \cos[r(t)] - y(t) + \frac{K}{\omega_{c2}} \frac{d \cos[r(t)]}{dt} \right\} \end{cases} \quad (1.14)$$

En dérivant une seconde fois le premier terme de l'équation (1.14), on obtient :

$$\frac{d^2 r(t)}{dt^2} = -g \frac{dy(t)}{dt} \quad (1.15)$$

Puis en injectant le deuxième terme de l'équation (1.14) dans l'équation (1.15) :

$$\frac{d^2 r(t)}{dt^2} = -g \omega_{c1} \left\{ K \cos[r(t)] - \frac{K}{\omega_{c2}} \sin[r(t)] \frac{dr(t)}{dt} - y(t) \right\} \quad (1.16)$$

Sachant par (1.14) que $y(t) = \frac{1}{g} \left(\omega_r - \omega_i - \frac{dr(t)}{dt} \right)$, l'équation différentielle de la PLL analogique s'écrit finalement :

$$\frac{d^2 r(t)}{dt^2} + \left(\omega_{c1} - K \cdot g \frac{\omega_{c1}}{\omega_{c2}} \sin[r(t)] \right) \frac{dr(t)}{dt} + g \omega_{c1} K \cos[r(t)] = \omega_{c1} (\omega_r - \omega_i) \quad (1.17)$$

Cette équation différentielle a été obtenue en prenant une sortie de comparateur de type sinusoïdal. Pour généraliser, il suffit de définir une fonction ψ (remplaçant ε), et donnant la différence de phase à la sortie du comparateur quelque soit la forme des signaux aux entrées de celui-ci. Cette fonction détermine la linéarité de l'équation différentielle régissant le système et le terme $\frac{d\psi[r(t)]}{dr(t)}$ est appelé sensibilité du comparateur. Dans ce cas, l'équation différentielle de la PLL analogique est donnée par :

$$\frac{d^2 r(t)}{dt^2} + \left(\omega_{c1} + g \frac{\omega_{c1}}{\omega_{c2}} \frac{d\psi[r(t)]}{dr(t)} \right) \frac{dr(t)}{dt} + g \omega_{c1} \psi[r(t)] = \omega_{c1} (\omega_r - \omega_i) \quad (1.18)$$

1.6. Dynamique normalisée

Même si la démarche consistant à normaliser l'équation (1.17) n'est pas nécessairement obligatoire, la normalisation est souvent utilisée [36-40, 44-46] afin de donner une forme plus classique à l'équation et de disposer d'une échelle plus avantageuse. Pour cela, on définit un temps réduit t_p en posant :

$$t_p = t \sqrt{g \omega_{c1} K} \quad (1.19)$$

L'équation (1.17) devient alors :

$$\frac{d^2 r(t_p)}{dt_p^2} + \frac{1}{\sqrt{g \omega_{c1} K}} \left(\omega_{c1} - K \cdot g \frac{\omega_{c1}}{\omega_{c2}} \sin[r(t_p)] \right) \frac{dr(t_p)}{dt_p} + \cos[r(t_p)] = \frac{\omega_r - \omega_i}{gK} \quad (1.20)$$

On peut maintenant étudier cette équation différentielle non linéaire à l'aide du plan de phase, en posant $x_1 = r(t)$ et $x_2 = \frac{dr(t)}{dt_p}$ puis en étudiant le système différentiel pour $t_p > 0$.

$$\begin{cases} \frac{dx_1}{dt_p} = x_2 \\ \frac{dx_2}{dt_p} = -\frac{1}{\sqrt{g\omega_{c1}K}} \left(\omega_{c1} - K \cdot g \frac{\omega_{c1}}{\omega_{c2}} \sin x_1 \right) \cdot x_2 - \cos x_1 + \frac{\omega_r - \omega_i}{gK} \end{cases} \quad (1.21)$$

Ce système d'équations fait apparaître les trois paramètres en fonction desquels l'évolution du système se fera. Ces paramètres $P_1 = \sqrt{\frac{\omega_{c1}}{gK}}$, $P_2 = \frac{\sqrt{gK\omega_{c1}}}{\omega_{c2}}$ et $P_3 = \frac{\omega_r - \omega_i}{gK}$ montrent le fait que pour stabiliser une PLL analogique et faire en sorte qu'elle synchronise, il faut un bon calibrage du filtre et veiller à ce que l'on soit dans la plage de verrouillage (écart de fréquence entre les signaux de référence et du VCO).

$$\begin{cases} \frac{dx_1}{dt_p} = x_2 \\ \frac{dx_2}{dt_p} = (-P_1 + P_2 \sin x_1) \cdot x_2 - \cos(x_1) + P_3 \end{cases} \quad (1.22)$$

Il est à noter qu'un modèle polynomial de PLL analogique est possible [41]. Pour se faire, on fait l'hypothèse que l'angle x_1 de l'équation (1.21) est petit :

$$\sin(x_1) \approx x_1 \text{ et } \cos(x_1) \approx 1 - \frac{x_1^2}{2}$$

Alors l'équation (1.21) devient alors

$$\frac{d^2 x_1}{dt_p^2} + P_1 \frac{dx_1}{dt_p} = \frac{P_2}{2} \frac{d(x_1^2)}{dt_p} + \frac{x_1^2}{2} + P_3 - 1 \quad (1.23)$$

Nous allons cependant analyser dans la section suivante, les trajectoires du système non linéaire de l'équation (1.22) afin de faire une analyse plus étendue.

1.7. Étude des trajectoires dans le plan de phase et paramétrage du système

Le plan de phase est construit en résolvant numériquement le système d'équations (1.22). A chaque instant t_p , un point de coordonnées $(x_1(t_p), x_2(t_p))$ est placé dans le **plan de phase**. Le dessin d'une solution $(x_1(t_p), x_2(t_p))$ pour $t_p > 0$, partant d'une **condition initiale** (condition fixée sur x_1 et x_2 : $x_1(0+)$ et $x_2(0+)$), s'appelle une **trajectoire** et elle montre le comportement dynamique du système. Cette trajectoire est unique à conditions initiales fixées¹. Plusieurs trajectoires peuvent ainsi être tracées, chacune correspondant à un couple de conditions initiales $x_1(0+)$ et $x_2(0+)$. De telles trajectoires sont tracées en exemple à la Fig. 1.3, où le paramètre P_2 est négligé. En effet, la plupart du temps, on a en pratique : $R_1 \gg R_2$.

$$P_1 = 0.5 \text{ et } P_3 = 0.7$$

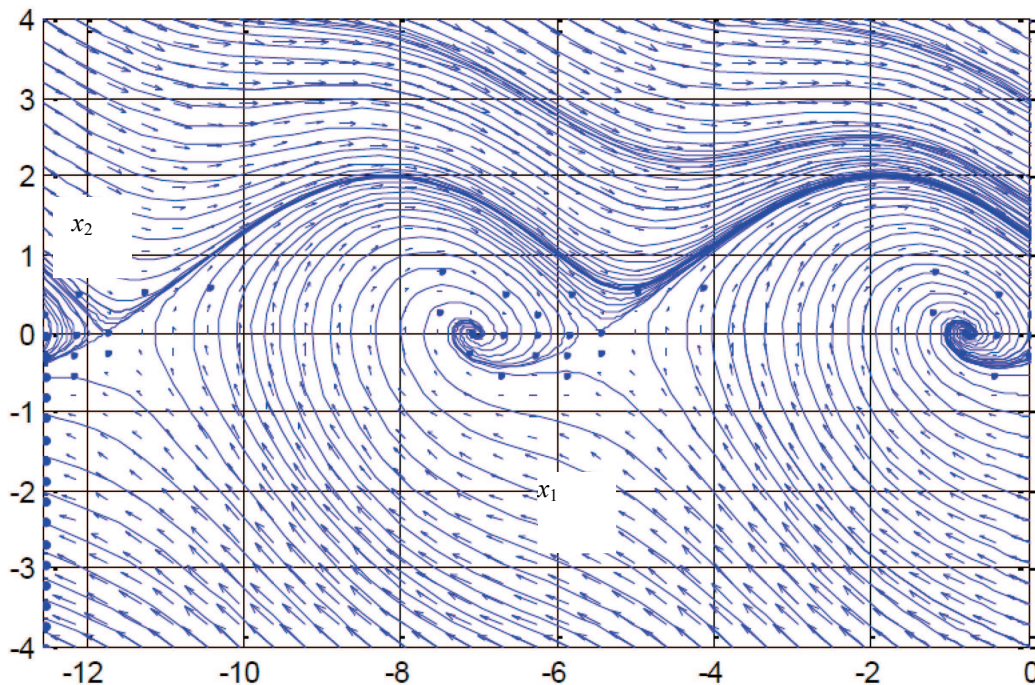


Fig. 1.3. – Plan de phase de la solution de l'équation (1.22)

¹Les conditions initiales sont choisies de sorte à former une grille en découpant l'axe x_1 de -11 à 0 avec un pas de 0.4 et en découpant l'axe x_2 de -4 à 4 avec un pas constant de 0.25.

Les courbes de la Fig. 1.3 (et des autres figures de cette section) ont été tracées sous MATLAB [85] en utilisant les fonctions de résolutions d'équations différentielles (ODE). Le code correspondant est donné en annexe A1.

1.7.1. Positions d'équilibre

Les positions d'équilibre sont données par la relation :

$$\begin{cases} \frac{dx_1}{dt_p} = 0 \\ \frac{dx_2}{dt_p} = 0 \end{cases} \quad \text{soit } x_2 = 0 \text{ et } \cos(x_1) = P_3 \quad (1.24)$$

Ces positions d'équilibre sont toutes situées sur l'axe des abscisses dans le plan de phase ($x_2 = 0$).

1.7.2. Analyse des portraits de phase

On définit les termes suivants :

Acquisition : Phénomène permettant à la PLL de capturer le signal d'entrée rapidement (sans saut de cycle), ou lentement (avec saut de cycle) puis de se verrouiller.

Zone de capture rapide (lock-in) : Ensemble des points (domaine du plan de phase) à partir desquels une trajectoire de phase tend vers un point d'équilibre stable sans saut de cycle. La variation de phase à partir de l'instant initial est inférieure à 2π en valeur absolue. La capture d'un signal d'entrée dont la trajectoire de phase passe par l'un de ces points entraîne toujours le verrouillage.

Limite de capture lente (pull-in) : valeur maximale de pulsation ou de fréquence du signal d'entrée de la PLL pour laquelle toutes les trajectoires de phase tendent, après éventuellement un ou plusieurs sauts de cycles, vers un point d'équilibre stable. L'acquisition se produit donc toujours.

On peut distinguer, à la suite d'essais successifs pendant lesquels on fait varier P_3 (P_1 étant fixé à 0.223), trois types de portraits de phase :

Premier cas : $P_3 \geq 0.94$ (Fig. 1.4)

Dans ce cas, il n'existe aucun point d'équilibre. Toutes les trajectoires de phases tendent vers un cycle limite (stable). L'acquisition n'a jamais lieu et la PLL ne verrouille jamais. Le cycle limite est dit stable car toutes les trajectoires de phase démarrant au dessus ou en dessous de ce cycle limite tendent vers ce dernier.

$$P_1 = 0.223 \text{ et } P_3 = 1$$

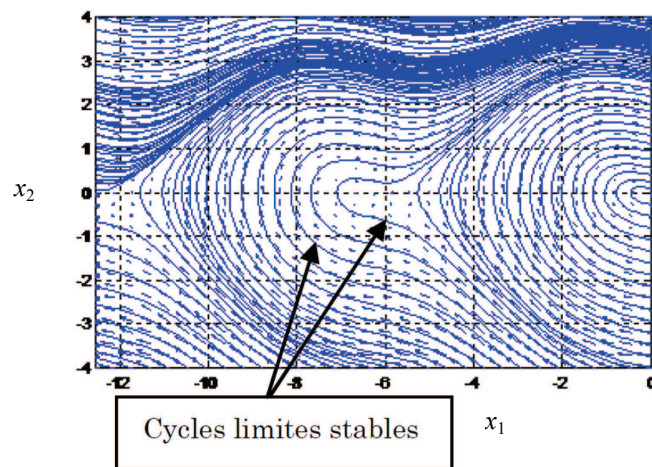


Fig. 1.4. – Plan de phase de la solution de l'équation (1.22)

Deuxième cas : $0.94 \geq P_3 > 0$ (Fig. 1.5)

Dans ce cas, certaines trajectoires de phase tendent vers un cycle limite stable mais d'autres tendent vers un point d'équilibre stable.

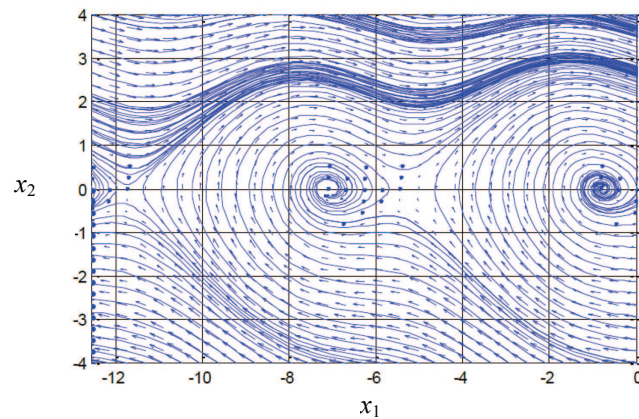


Fig. 1.5. – Plan de phase de la solution de l'équation (1.22) pour $P_3 = 0.8$.

Troisième cas: Pour $P_3 = 0$ (ou P_3 très proche de 0)

Dans ce cas, il y a toujours verrouillage avec éventuellement saut de cycles, il n'y a plus de cycles limites.

Lors du calibrage d'une PLL analogique, en fonction des contraintes du cahier des charges, on peut être amené :

- soit à fixer le paramètre P_1 comme cela est présenté dans cette section, sachant que la seule exigence théorique est qu'il satisfasse le critère du rejet de la composante *ripple*. On fait ensuite varier le paramètre P_3 pour observer le comportement et déterminer la plage de verrouillage du système.
- soit à fixer le paramètre P_3 si des limitations fréquentielles sont imposées et à faire varier P_1 (dans une plage permettant d'éliminer la composante *ripple*) en vue d'obtenir les performances recherchées (en termes par exemple de marge de module et marge de gain).

1.8. Conclusion

Ce chapitre a développé l'étude d'une PLL analogique et a permis d'introduire les notions de base intervenant dans la compréhension des PLLs. Il est possible de s'inspirer du développement fait dans cette partie consacrée à l'étude dynamique d'une PLL analogique d'ordre deux pour étudier d'autres PLLs d'ordre supérieur. Par exemple, un semblable développement a été fait entre autre par Harb, Monteiro et Piqueira [37-40, 42-43] pour mettre en évidence les bifurcations et le chaos dans une PLL analogique d'ordre trois. D'autre part, Tanaka a montré, sur la base d'un développement similaire qu'un couplage de deux PLLs d'ordre 2 pouvait générer du chaos sous certaines conditions, en effectuant une transformation ramenant l'ordre 4 du système à un ordre 3 [44]. En effet, dans un système continu, il ne peut pas y avoir de chaos si l'ordre du système est inférieur à 3.

Après avoir traité des dispositifs uniquement analogiques et acquis les notions essentielles de fonctionnement d'une PLL, nous allons nous intéresser dans le chapitre suivant à la boucle à verrouillage de phase digitale (ADPLL) dont l'utilisation est extrêmement répandue. Les principales approches qui vont suivre s'appuient sur des

méthodes d'analyse en régime linéaire. En effet, bien que les systèmes physiques ne soient jamais linéaires, on peut s'en approcher sous certaines conditions.

Chapitre 2 :

La boucle à verrouillage de phase digitale

2.1. Introduction

Il existe plusieurs types de PLLs qui ne sont malheureusement pas rigoureusement distinguées dans la littérature. Par exemple, Abramovitch affirme dans [64] qu'une PLL digitale peut être constituée uniquement de composants analogiques à l'exception du détecteur de phase (digital). Cependant, ces derniers types de PLLs ne sont pas considérés par d'autres auteurs [65] comme des PLLs digitales dans le sens où les signaux issus de cette PLL ne permettent pas de constituer un système de données échantillonnées. De ce dernier point de vue, La PLL en question est juste une PLL analogique implémentée avec un détecteur de phase digital dont la sortie peut être vue comme une tension continue utilisée pour piloter un filtre analogique. Pour éviter ce genre de confusions, nous allons distinguer 3 types de PLLs:

Les PLLs analogiques dont tous les composants sont analogiques, les PLLs semi-numériques, digitales ou mixtes dont certains éléments sont analogiques et d'autres numériques en enfin les PLLs entièrement numériques (ADPLLs). Bien que Les PLLs analogiques soient encore largement utilisées, de nombreuses raisons amènent à préférer l'utilisation des circuits digitaux pour implémenter des PLLs. Les ADPLLs¹ présentent en effet plusieurs avantages sur leurs homologues analogiques dont la facilité d'intégration et de vérification. Les ADPLLs permettent en outre de réduire de nombreux problèmes associés aux PLLs analogiques. Par exemple, les PLLs analogiques souffrent de la sensibilité du VCO aux variations de température et d'alimentation, d'où la nécessité d'effectuer un étalonnage initial et de fréquents ajustements alors que les ADPLLs sont comparativement moins sensibles à ce genres de problèmes [47]. Un autre point est le fait que les multiplicateurs analogiques souvent utilisés dans les PLLs analogiques sont sensibles aux fluctuations des signaux d'entrées [28,52], plus coûteux

¹ Boucle à verrouillage de phase dont tous les composants sont digitaux

et ne sont pas réalisables avec des composants numériques. Le détecteur de phase digital (DPD) utilisé par les PLLs mixtes et les ADPLLs quant à lui, du fait qu'il fonctionne sur la détection des fronts des signaux d'entrée, est plus simple à mettre en œuvre moyennant l'utilisation de bascules et/ou de composants logiques. On dispose de plus de latitude pour réaliser des filtres numériques (basée sur les valeurs de coefficients d'une équation aux différences) qu'analogiques (basée sur les valeurs de résistances et de capacités). Le manque de précision qui peut subsister dans le choix des paramètres de la PLL analogique rend l'acquisition souvent lente et peu fiable tandis que les ADPLLs verrouillent plus vite [48]. Enfin, il est possible de réaliser des traitements plus complexes en numérique qu'en analogique, le tout pour un gain plus intéressant en surface de silicium.

Dans cette partie, nous allons introduire l'ADPLL, proposer des exemples d'architecture existants, donner un exemple de conception d'une ADPLL d'ordre deux et enfin, décrire la PLL auto-échantillonnée digitale autour de laquelle s'articule le projet HODISS.



Fig. 2.1 – Schéma de base de l'ADPLL.

2.2. Classification des ADPLLs

On peut distinguer plusieurs implémentations d'ADPLLs en fonction de la mécanique du détecteur de phase. D'après les classifications effectuées dans la littérature, de cette mécanique peuvent résulter deux processus d'échantillonnage (uniforme ou non). Les ADPLLs dont celles fonctionnant avec un comparateur XOR ou à bascules (nous donnons des exemples dans les sections suivantes) sont considérées comme non uniformément échantillonnées. En effet, dans ces dernières ADPLLs, l'erreur de phase est déduite de la durée entre les instants d'activation et de remise à zéro des bascules. Ces dernières bascules fonctionnant sur les fronts des signaux d'entrée dont l'horloge locale qui a une fréquence variable en régime transitoire. Seules les ADPLLs échantillonnées à la fréquence de Nyquist [53-54] (NR-DPLLs pour Nyquist Rate Digital Phase-Locked

Loops) sont considérées comme étant uniformément échantillonnées [47]. De ce fait, la terminologie utilisée dans ce rapport diffère quelque peu de celle utilisée dans la littérature en ce sens que nous entendons par échantillonnage non uniforme [94], non seulement le fait que le filtre numérique ne soit pas échantillonné avec une horloge régulière, mais que cette horloge diffère de celle donnant la disponibilité de l'erreur de phase issue du détecteur. Nous reviendrons sur cette nuance dans la section traitant des SS-ADPLLs.

2.2.1. L'ADPLL avec un comparateur XOR

Les ADPLLs avec un comparateur de phase XOR dont un schéma synoptique est donné en Fig. 2.2 sont très utilisées à cause de la simplicité du détecteur. La réponse d'un détecteur de phase XOR donnée en Fig. 2.4, a une plage de fonctionnement linéaire plus large que celle d'un détecteur analogique.

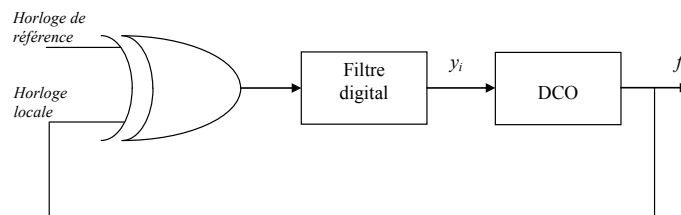


Fig. 2.2 – ADPLL avec un comparateur XOR classique

Le détecteur de phase XOR fonctionne de sorte à générer des impulsions (signal d'erreur) sur les fronts montants et les fronts descendants des horloges de référence et de retour de boucle (Fig. 2.3). En fonction de la variation de la différence de phase des signaux d'entrée, donc de la largeur des impulsions du signal d'erreur en sortie du détecteur, un niveau DC noté y_i est fourni. Pour caractériser le détecteur, définissons par e_r la différence temporelle entre l'arrivée du front montant de l'horloge de référence (de période T_r) et celle de l'horloge locale (de période centrale T_i et de fréquence f_i). Si on désigne par ϕ_r la phase du signal de référence et par ϕ_i celle de l'horloge locale, l'erreur de phase ϕ_e est donnée par :

$$\phi_e = \phi_i - \phi_r = \frac{e_r}{T_r} 2\pi \quad (2.1)$$

Le niveau DC correspondant à la moyenne du signal d'erreur en sortie du détecteur peut être exprimé par :

$$y_i = K_{dpd} \cdot \phi_e \quad (2.2)$$

ou K_{dpd} est le gain du détecteur :

$$K_{dpd} = \frac{y_{i\max}}{\pi} \quad (2.3)$$

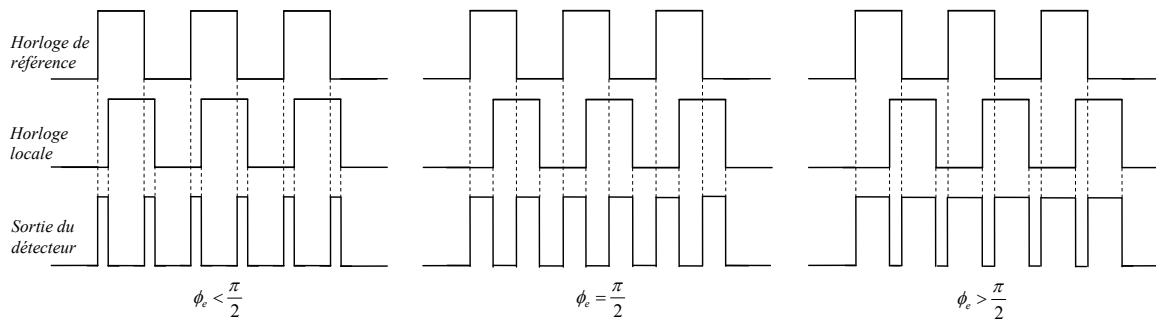


Fig. 2.3 – Détection de l'erreur de phase avec une porte logique XOR

Lorsque le un rapport cyclique¹ entre le signal de référence et celui de l'horloge locale est de 50%, on dit que l'ADPLL est verrouillée [55]. Le déphasage ϕ_e est alors de $\frac{\pi}{2}$ comme indiqué sur la Fig. 2.3, ce qui signifie que le niveau DC vaut :

$$y_i = \frac{y_{i\max}}{2} \quad (2.4)$$

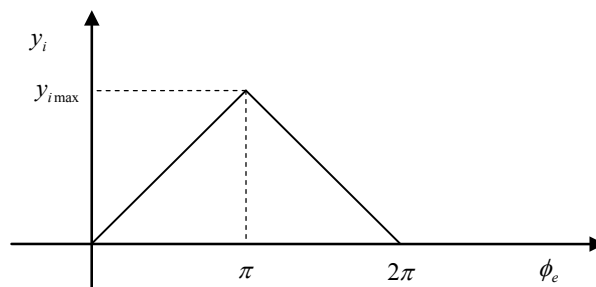


Fig. 2.4 – Caractéristique du détecteur XOR

Si le déphasage ϕ_e s'écarte de $\frac{\pi}{2}$, le rapport cyclique du signal de sortie du détecteur change proportionnellement à la différence de phase mesurée, entraînant de ce fait le

¹Rapport entre le temps à l'état haut dans une période et la valeur de cette période

changement de sa valeur moyenne y_i . La Fig. 2.5 donne la fréquence de sortie de l'horloge locale en fonction du signal de contrôle y_i .

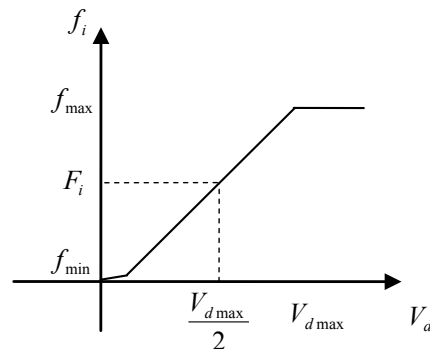


Fig. 2.5 – Fréquence de l'horloge locale en fonction du code de contrôle y_i

Par exemple, dans son étude, Greer a utilisé une porte XOR comme détecteur de phase, un compteur *modulo K* comme filtre numérique et un autre compteur pouvant incrémenter et décrétement (compteur et décompteur I/D) comme oscillateur à commande digitale (DCO). La sortie du détecteur de phase contrôle les opérations d'un filtre digital consistant en un compteur réversible, c'est-à-dire un compteur qui permet de compter ou décompter selon le niveau logique y_i appliqué à une broche de commande appelée sens de comptage (*Up/Down*). Le schéma simplifié (sans diviseur de fréquence sur le signal de retour de la boucle) de l'ADPLL conçu par Greer est donné en Fig. 2.6. Les compteurs et décompteurs du filtre digital [90] sont tous les deux pilotés par une horloge de fréquence $f_e = MF_i$, où M est un entier. La sortie H du filtre est connectée à l'entrée d'incrémentement (*INC*) du DCO et génère une impulsion à la fin d'un cycle 'haut' en sortie du détecteur pendant que la sortie B du filtre, connectée à l'entrée de décrémentation (*DEC*) du DCO, génère une impulsion à la fin d'un cycle 'bas' en sortie du détecteur.

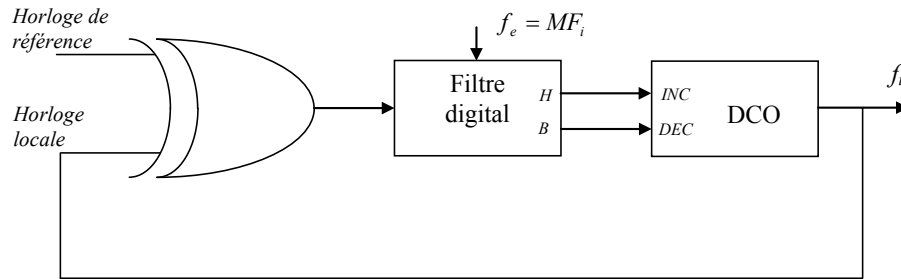


Fig. 2.6 –Schéma simplifié de l'ADPLL XOR de Greer

Une impulsion appliquée à l'entrée *INC* ajoute 1/2 cycle au signal de sortie du DCO tandis qu'une impulsion sur l'entrée *DEC* retire 1/2 cycle. La fréquence de sortie est donnée par:

$$f_i = F_i + \frac{1}{2} [y_i \cdot M \cdot F_i / K] \quad (2.5)$$

2.2.2. Les ADPLLs avec comparateurs à bascules

2.2.2.1. Principe

Ce genre d'ADPLLs a été proposé dans la littérature par bon nombre d'auteurs [55-60]. La sortie du détecteur est souvent utilisée pour commander la circulation du courant dans un circuit de pompe de charge (ensemble composé de deux transistors) dans les PLLs mixtes¹. Le détecteur (Fig. 2.7) est réalisé à partir d'une porte NAND et de deux bascules D dont on peut contrôler la réinitialisation (RST) en fonction des arrivées des fronts montants de la référence et de la sortie de l'oscillateur commandé.

¹ Boucle à verrouillage de phase dont certains blocs sont analogiques et d'autres numériques, ces boucles sont souvent désignées dans la littérature par l'acronyme DPLLs (Digital Phase-Locked Loops).

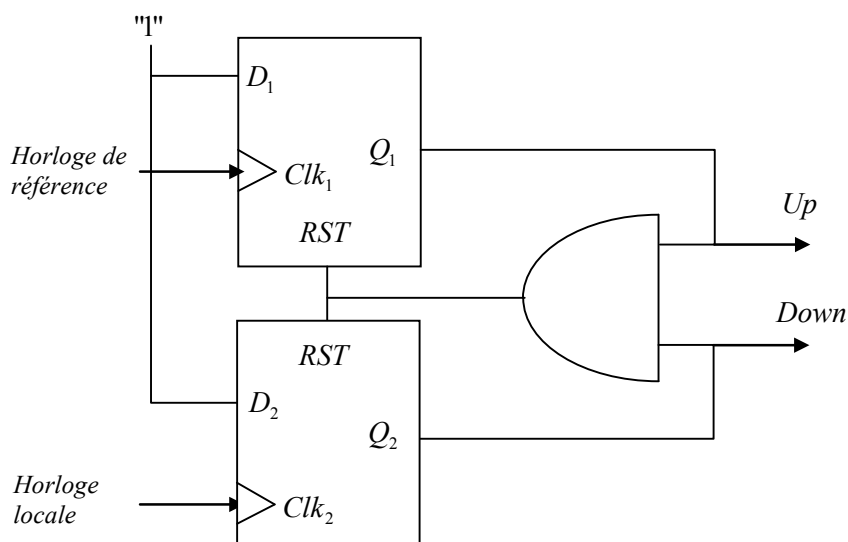


Fig. 2.7 – Schéma d'un comparateur de phase et de fréquence à bascules D

Avec ce type de détecteur, on n'est pas sensible au rapport cyclique des signaux de commande parce que les bascules ne sont sensibles qu'aux fronts montants. Supposons qu'au départ, les deux bascules sont initialisées à zéro ($Q_1 = Q_2 = "0"$). L'arrivée d'un premier front montant de l'horloge de référence (resp. de l'horloge locale) entraîne l'activation de la bascule correspondante et le passage à "1" de Q_1 (resp. le passage à "1" de Q_2). Il se passe la même chose au prochain front montant détecté sauf si la condition $Q_1 = Q_2 = "1"$ est vérifiée. La porte NAND sort alors "1", remettant à zéro les deux bascules. Les chronogrammes du détecteur sont donnés en Fig.2.8-2.9 et mettent en évidence deux modes de fonctionnement permettant de corriger à la fois la phase et la fréquence. Trois conditions sont possibles pour l'erreur de phase :

1) Si l'horloge de référence est en avance, la sortie Up du détecteur génère des impulsions dont la durée est égale à la différence de temps e_r entre les arrivées des fronts montants des deux horloges de commande. Cela signifie qu'un signal de commande proportionnel à l'erreur sera envoyé après filtrage au DCO pour en augmenter la fréquence. La sortie $Down$ correspond à des impulsions très brèves.

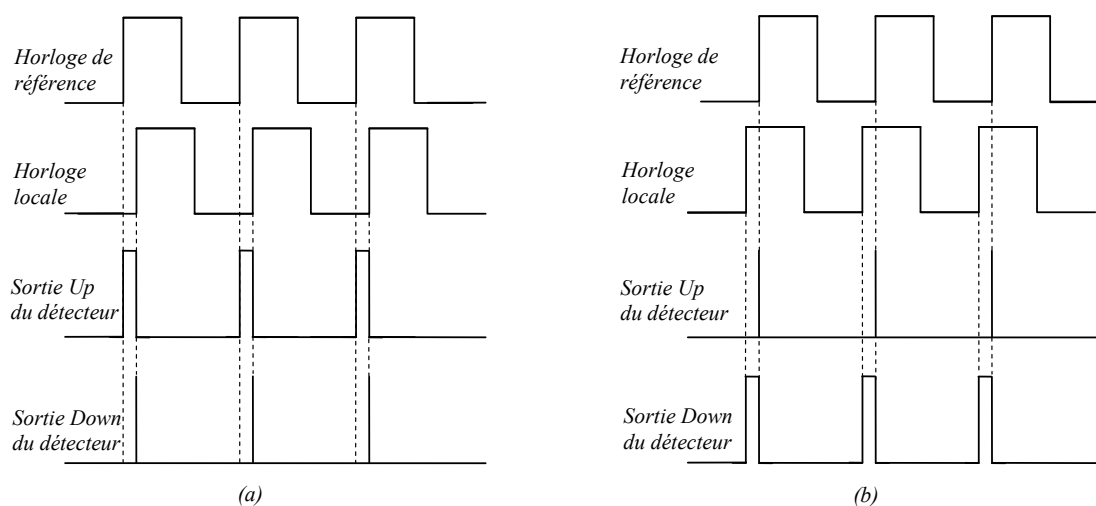


Fig. 2.8 – Fonctionnement en comparateur de phase du détecteur à bascules : (a) Horloge de référence en avance par rapport à l'horloge locale, (b) Horloge de référence en retard par rapport à l'horloge locale.

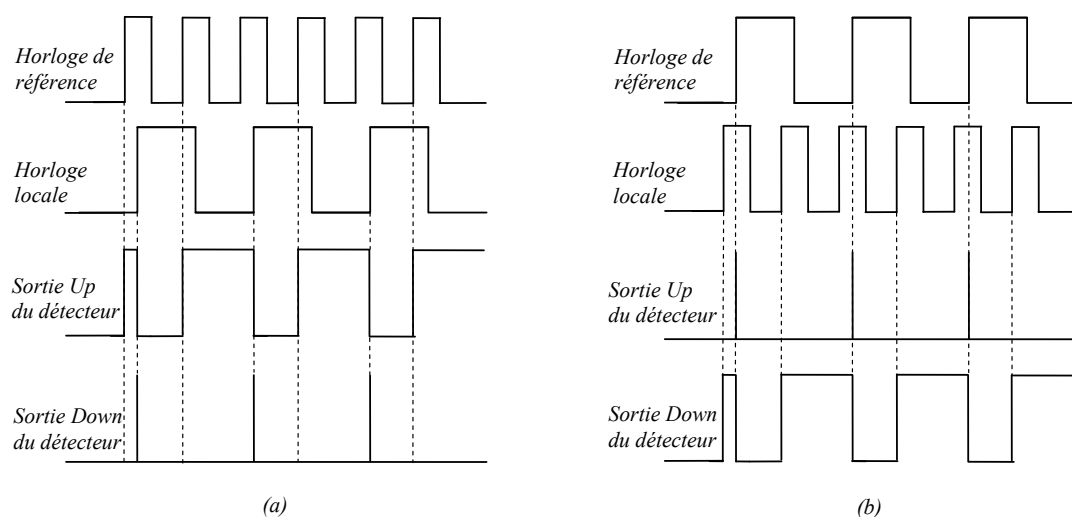


Fig. 2.9 – Fonctionnement en comparateur de fréquence du détecteur à bascules : (a) Fréquence de l'horloge de référence supérieure à celle de l'horloge locale, (b) Fréquence de l'horloge de référence inférieure à celle de l'horloge locale.

2) Si au contraire, l'horloge de référence est en retard, c'est la sortie *Down* du détecteur qui générera les impulsions de durée e_r . Un signal proportionnel à l'erreur sera envoyé, après filtrage au DCO pour en diminuer la fréquence. La sortie *Up* correspond à de brèves impulsions.

3) la troisième situation correspond au cas où les deux signaux de commande sont synchrones. Les signaux *Up* et *Down* du détecteur sont tous les deux des impulsions

très brèves. Lorsque le déphasage est très proche de zéro, le comparateur présente une zone morte (visible sur la caractéristique de la Fig. 2.10) due au retard induit par la porte logique AND qui gère la remise à zéro des bascules.

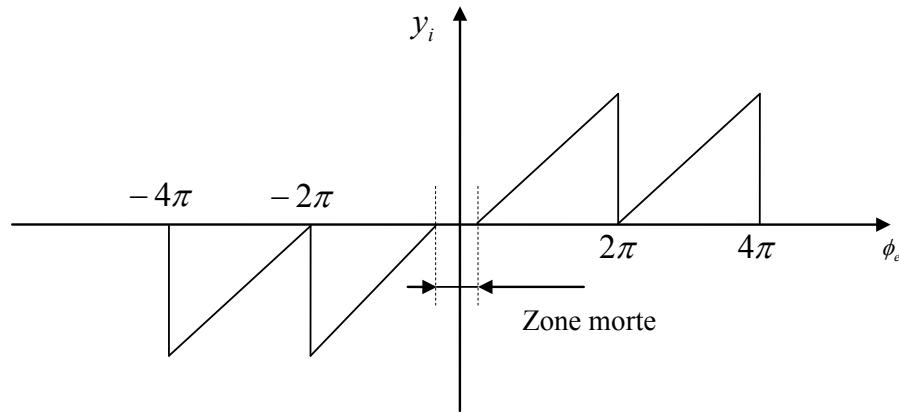


Fig. 2.10 – Caractéristique de détecteur de phase et de fréquence à bascules D.

Dans les PLLs mixtes, extrêmement populaires, les signaux *Up* et *Down* du détecteur de phase commandent à une pompe de charge d'injecter du courant ou de le retirer du filtre proportionnellement à l'écart e_r et donc proportionnellement à l'erreur de phase. Ce même principe peut être exploité pour la conception d'une ADPLL. Il suffit pour cela d'utiliser un convertisseur de délai (TDC pour Time to Digital Converter) pour convertir la valeur de l'écart temporelle en une valeur digitale sur les sorties *Up* (resp. *Down*). Ces valeurs pourront ensuite être filtrées par un filtre digital, lui-même générant un code permettant de faire varier la fréquence du DCO. Le schéma de cet exemple d'ADPLL est donné en Fig 2.11. Dans la section suivante, un modèle Simulink [100, 106] est proposé en vue de d'illustrer le fonctionnement de l'ADPLL.

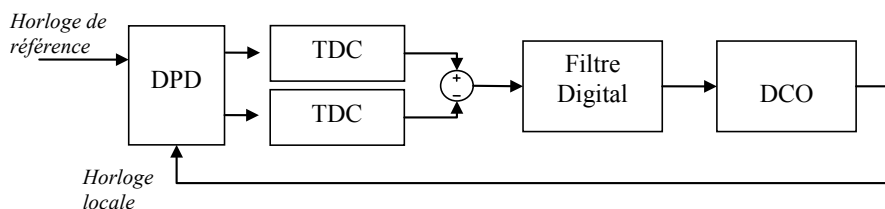


Fig. 2.11 – Modèle d'ADPLL avec un DPD à bascules D.

2.2.2.2. Exemple de conception d'une ADPLL, modèle Simulink

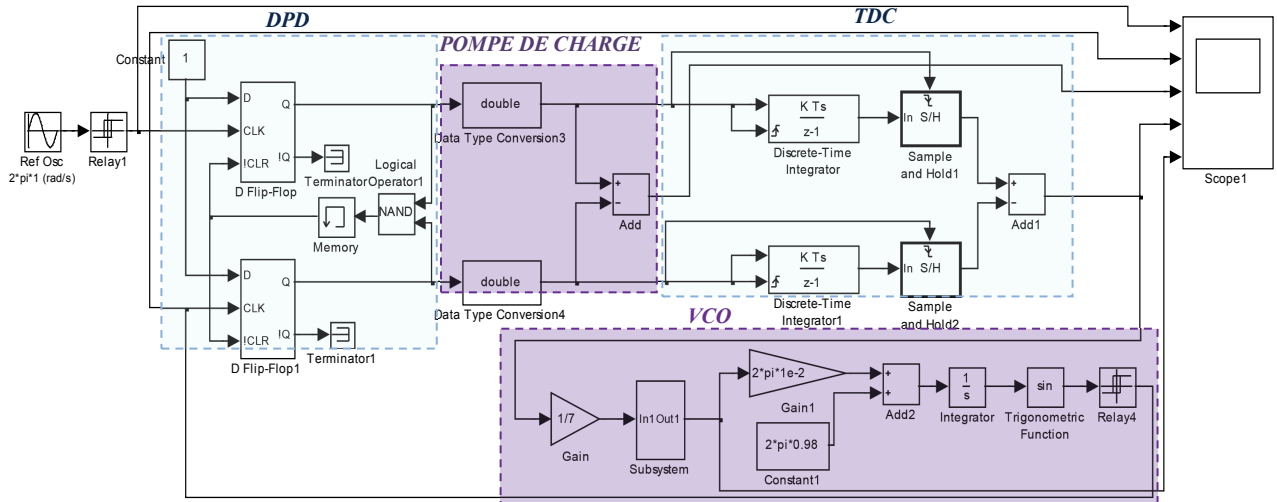


Fig. 2.12 – Modèle Simulink d'une ADPLL avec un DPD à bascules D.

Cette partie présente, à but uniquement illustratif, les résultats obtenus à l'aide d'un modèle Simulink d'ADPLL d'ordre deux, pour l'architecture proposée dans [55-60]. Par souci de simplicité, le fonctionnement du TDC et du DCO sont idéalisés, ces deux blocs étant modélisés à l'aide d'intégrateurs discrets purs, échantillonnés uniformément. Les paramètres utilisés pour simuler l'ADPLL sont normalisés, c'est-à-dire que nous considérons une fréquence unité pour le signal de référence. Les valeurs de ces paramètres sont données dans le tableau 2.1. Le filtre utilisé est un correcteur de premier ordre dont la fonction de transfert est donnée par :

$$H(z) = \frac{A + B \cdot z^{-1}}{1 - z^{-1}} \quad (2.6)$$

Saturation du TDC	$K = 1$ [Code_V, Niveau logique]
Période du TDC (Intégration)	$\tau = 0.01$ [s]
Gain du DCO	$g = 0.01$ [Hz/Code_V]
Fréquence centrale du DCO	$F_i = 0.98$ [Hz]

Tableau 2.1 – Paramètre du modèle Simulink de l'ADPLL avec un DPD à bascules D.

Pour le choix de coefficients du filtre $A = 0.2282$ et $B = -0.2244$, l'ADPLL verrouille comme l'illustre la Fig. 2.13.

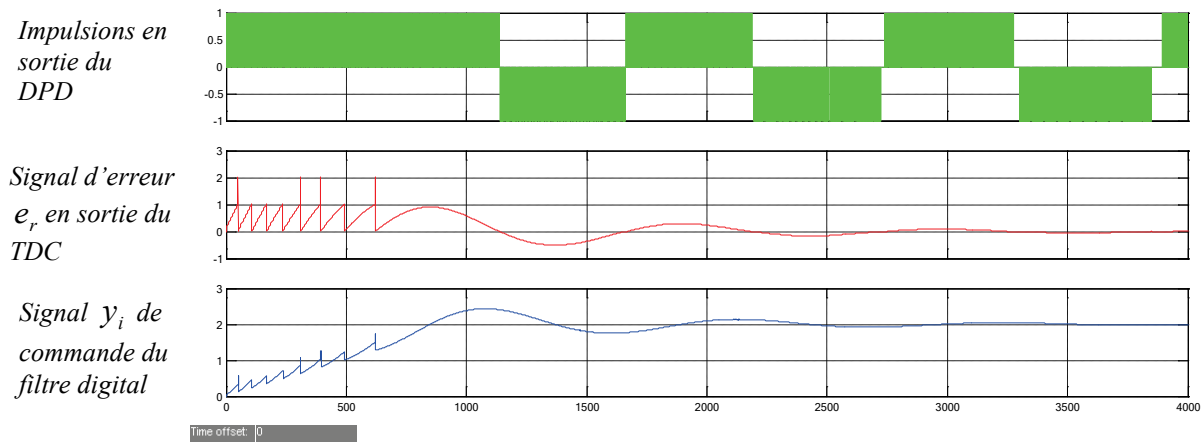


Fig. 2.13 – Simulation transitoire d'une ADPLL avec un DPD à bascules D pour un couple de coefficients $A = 0.2282$ et $B = -0.2244$. En vert : Impulsions de durée proportionnelle à l'erreur de phase en sortie du DPD, En rouge : Signal d'erreur temporelle en sortie du TDC. En bleu : Signal de commande du DCO.

Dans la section suivante, nous allons traiter un aspect plus réaliste de la mise en œuvre d'une ADPLL et la solution adoptée dans le projet pour l'échantillonnage du filtre numérique.

2.3. L'ADPLL auto-échantillonnée (SS-ADPLL)

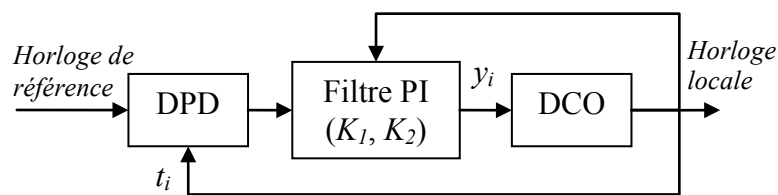


Fig. 2.14 – Schéma de base de la SS-ADPLL.

Le schéma des SS-ADPLLs développées durant le projet HODISS est donné en Fig. 2.14. Sa particularité est que le filtre de boucle est cadencé par la sortie de l'horloge locale.

2.3.1. Le DPD mis en œuvre dans le projet HODISS

Nous allons ici décrire le fonctionnement du détecteur de phase et de fréquence N-bit ($N=16$) utilisé dans le projet. Il est composé d'un détecteur de phase et de fréquence de type bang-bang (BB-DPD) [19, 20, 22, 24, 102] et d'un convertisseur TDC.

2.3.1.1. Le détecteur bang-bang (BB-DPD)

Un détecteur de phase et de fréquence bang-bang produit une information sur le signe de l'erreur de phase entre deux signaux. Les signaux sont représentés par des événements correspondants aux instants où l'on détecte les fronts montants des signaux (carrés) de référence et de l'horloge locale. Nous avons appelé les événements correspondants *Ref* et *Div* respectivement. Nous noterons également les instants où l'événement *Ref* apparaît t_r et ceux où l'événement *Div* apparaît t_i .

Concrètement, un détecteur de phase bang-bang détecte lequel des événements (*Ref* ou *Div*) est arrivé en premier au niveau de ses deux entrées. Sa sortie est un code binaire qui signifie soit que :

- L'événement *Ref* est arrivé en premier ou au contraire que c'est
- L'événement "*Div*" qui est arrivé en premier.

Le détecteur bang-bang fonctionne comme une machine à 4 états (Fig. 2.15), c'est à dire qu'il comprend 2 bits de mémoire que nous nommerons *Signe* et *Mesure*.

Le bit Signe : Sa valeur est nulle si la référence est en avance sur l'horloge locale. Dans le cas contraire, sa valeur vaut 1.

Le bit Mesure : Lorsque le détecteur démarre, le bit *Mesure* est mis à 0. Dès la détection d'un événement en entrée du détecteur, le bit *Mesure* se met à 1. Le bit *Signe* prend la valeur respectivement de 0 ou de 1 si l'événement est *Ref* ou *Div*.

Si *Mesure*=1 et que *Signe*=0, cela signifiant que l'événement arrivé en premier est *Ref*, la machine attend l'événement *Div*, ignorant tous les autres événements *Ref*. Lorsque l'événement *Div* arrive, le bit *Mesure* se remet à 0 tandis que le bit *Signe* reste à 0.

Dans la même logique, Si *Mesure*=1 et que *Signe*=1, cela signifiant que l'événement arrivé en premier est *Div*, la machine attend l'événement *Ref*, ignorant tous les autres

événements *Div*. Lorsque l'événement *Ref* arrive, le bit *Mesure* se remet à 0 tandis que le bit *Signe* reste à 1.

Le fonctionnement de l'automate à 4 états est décrit en Fig. 2.15. Les bits de haut représentent *Signe* et ceux de bas *Mesure*. Si le BB-DPD détecte le signe de l'erreur de phase, le TDC quant à lui mesure la valeur absolue de cette erreur par niveaux de quantification.

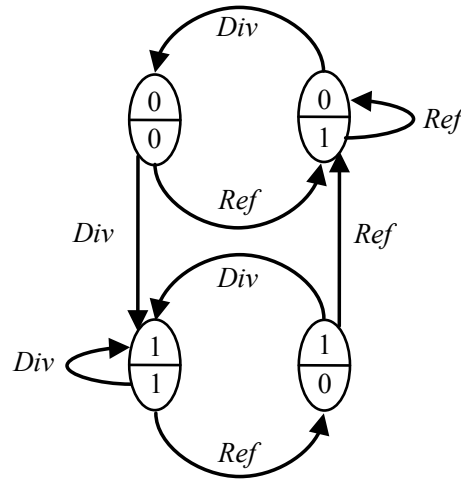


Fig. 2.15 – Automate à 4 états du détecteur de phase et de fréquence bang-bang.

Les signaux de sortie du TDC et du BB-DPD sont reçus par un bloc qui donnera à sa sortie un code correspondant à l'erreur de phase signée. L'opération à ce niveau consistera en une multiplication de l'erreur de phase absolue avec le signe de cette erreur (Fig. 2.16). Pour être plus précis, la sortie du codeur sera un arrondi correspondant au complément à deux de l'erreur de phase signé. Le principal avantage de ce type de détecteur comparé avec d'autres types de DPDs est le fait qu'il est possible d'obtenir une valeur de code très faible lorsque l'ADPLL est dans son état verrouillé.

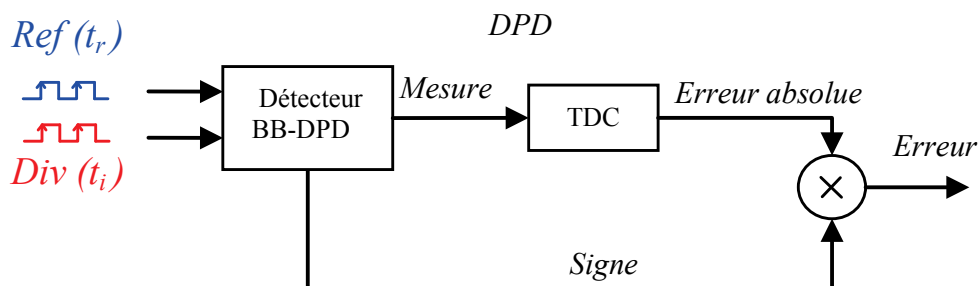


Fig. 2.16 – Schéma du détecteur de phase et de fréquence.

2.3.1.2. Fonctionnement du TDC

Une des techniques les plus populaires pour la mise en œuvre d'un TDC dans les circuits VLSI à haute fréquence consiste à propager le signal à travers des lignes de retard, et observer le délai de propagation durant l'intervalle de temps à mesurer. Ce principe est décrit par Levine dans [61] et le schéma correspondant est donné sur la Fig. 2.17. Ces TDC ont la caractéristique donnée à la Fig. 2.18(a). Souvent le TDC est conçu de sorte à avoir une saturation lorsque le délai est plus grand qu'une certaine valeur seuil correspondant au plus grand délai possible des lignes de retard. La caractéristique de TDC donné en Fig. 2.18(a) présente le cas où la quantification est linéaire. C'est le cas pour le DPD implémenté dans le projet HODISS montré en Fig. 2.19, à la seule différence qu'il intègre un effet bang-bang en zéro (+1 pour de faibles erreurs positives et -1 pour de faibles erreurs négatives). Cependant, il peut être intéressant d'attribuer aux intervalles de retards, des valeurs différentes de sorte à obtenir un TDC non-linéaire comme en Fig. 2.18(b). C'est le cas par exemple, lorsque la précision voulue est différente pour certaines gammes d'intervalle. Si on dispose d'une ligne de retard composée de K délais individuels et que la valeur du délai individuel est notée τ_i , alors la fonction donnée par le TDC est :

$$TDC_{out}(\tau) = \begin{cases} 0, & 0 < \tau < \tau_1 \\ i, & \sum_{j=1}^{j=i} \tau_j < \tau < \sum_{j=1}^{j=i+1} \tau_j \text{ et } 1 \leq i \leq K-1 \\ K, & \tau > \sum_{j=1}^{j=K} \tau_j \end{cases} \quad (2.7)$$

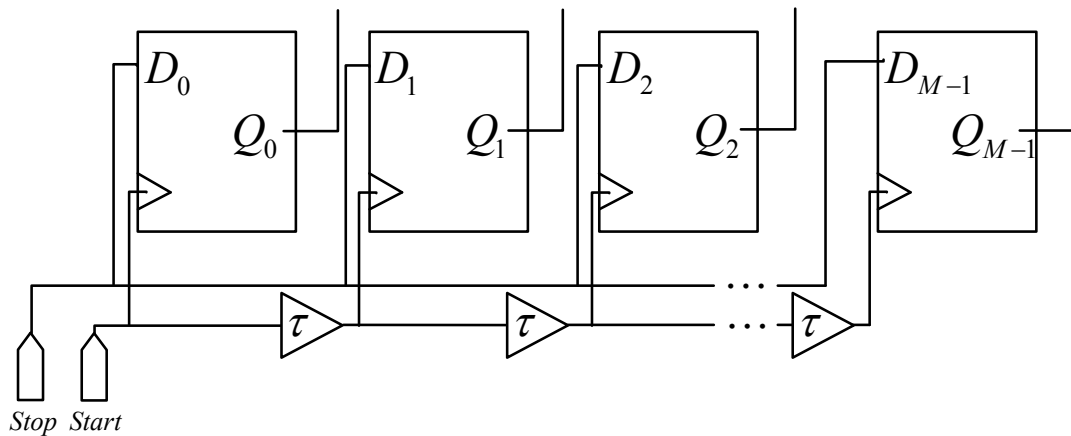


Fig. 2.17 – Une implémentation physique possible d'un TDC.

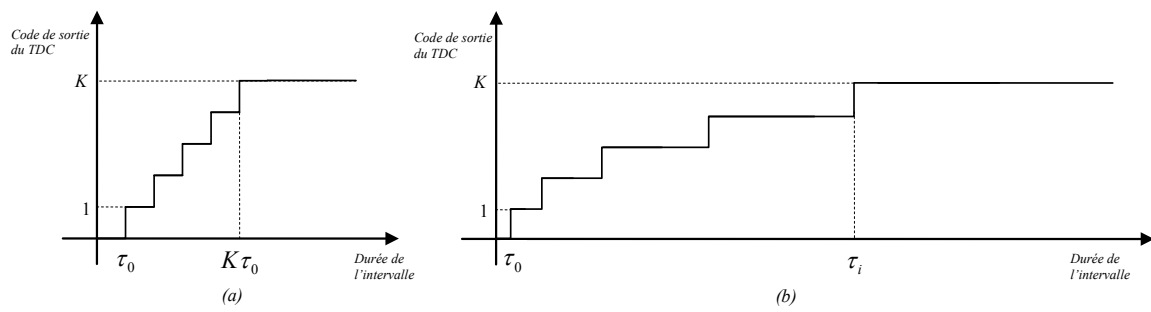


Fig. 2.18 – Caractéristiques d'un TDC : (a) linéaire, (b) non linéaire.

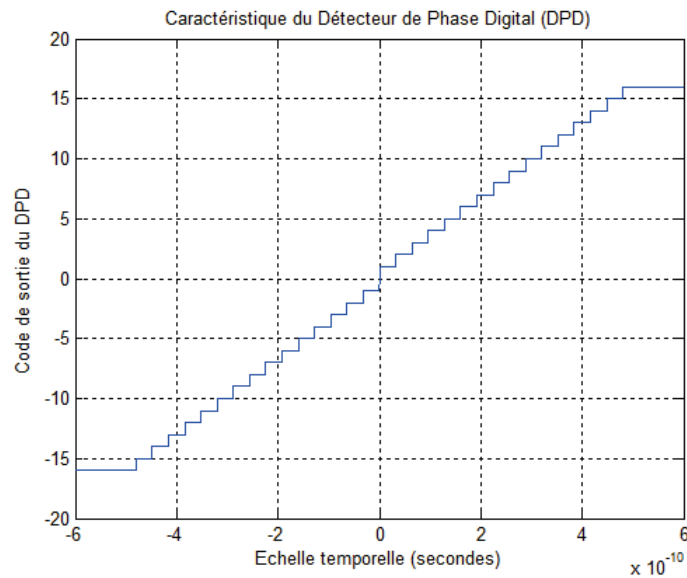


Fig. 2.19 – Caractéristique du DPD HODISS.

Le DPD va dans toute la suite, dans toute la modélisation théorique qui suivra, être supposé linéaire. On considèrera qu'il sort un code proportionnel à la valeur de l'erreur temporelle, c'est-à-dire le temps écoulé entre l'arrivée d'un front montant de l'horloge de référence et celui de l'horloge locale. Notons que cette description n'est valable que lorsque l'ADPLL est proche de la synchronisation : cependant, cette hypothèse n'est pas restrictive pour l'étude de la stabilité de l'état synchronisé, puisque nous nous intéressons seulement à de petites perturbations autour du point d'équilibre stable (état synchronisé). Le fait qu'en pratique, le DPD ait une caractéristique avec de la saturation, garantit que l'ADPLL se comporte comme une PLL bang-bang lorsque nous nous situons loin de la synchronisation et, par conséquent, a une large plage de verrouillage. Une description détaillée des blocs de construction du DPD peuvent être trouvés dans [29].

2.3.2. Le filtre proportionnel-intégral (PI)

Les systèmes asservis pouvant présenter des défauts, une précision insuffisante, une stabilité trop sensible aux perturbations (voire une instabilité), un temps de réaction trop lent, un dépassement trop important, au regard du cahier des charges, il est souvent nécessaire d'intégrer dans le système asservi un système correcteur dont l'objectif est d'améliorer un ou plusieurs de ces différents paramètres.

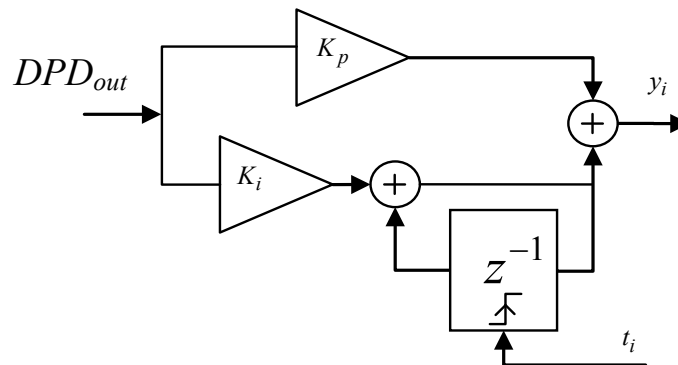


Fig. 2.20 – Schéma d'un filtre PI typique

Dans notre cas, le filtre proportionnel intégral (PI) est utilisé pour traiter le signal d'erreur en sortie du DPD et fournir le code digital qui alimentera le DCO. Dans certains types de PLLs notamment les PLLs analogiques, le filtre est paramétré pour supprimer le bruit et les composantes hautes fréquences du signal en son entrée. Dans notre projet, étant donné la nature complètement numérique et complexe de notre système, nous avons opté pour un filtre (correcteur) PI. Ceci d'une part pour un souci de simplicité en raison de son ordre (classe 1) et d'autre part du fait qu'il permet une amélioration très sensible de la précision (plus d'erreur de position). L'action de ce correcteur se fait sur les basses fréquences (Fig. 2.21). Il introduit un gain et une action intégrale. L'avantage majeur de ce correcteur est qu'il permet de synchroniser à la fois en phase et en fréquence [28] ; il annule l'erreur statique grâce à l'action intégrale. Son principal inconvénient est qu'il induit un retard de phase (de $\frac{\pi}{2}$) en basses fréquences à cause de l'action intégrale qui peut conduire à une instabilité. La fréquence à partir de laquelle ce dernier effet devient négligeable est fixé par le rapport $\frac{K_i}{K_p}$ (Sur la Fig. 2.20 qui montre

un filtre PI, K_p représente le gain de la branche proportionnelle et K_i celui de la branche intégrale). La nature de ce détecteur permet dans une certaine mesure de contourner le problème de la modularité de la phase en pratique. Les coefficients K_p et K_i déterminent toute la dynamique de la boucle comme nous l'avons vu pour le filtre passe bas au chapitre 1, à savoir la stabilité, la vitesse de convergence ou même la robustesse du système. La fonction de transfert du correcteur est donnée par :

$$H(z) = \frac{(K_p + K_i) - K_p z^{-1}}{1 - z^{-1}} \quad (2.8)$$

Les lieux de Bode du correcteur sont représentés en Fig. 2.21 pour $K_p = 0.7$ et $K_i = 0.07$ (couple de coefficients permettant de synchroniser l'ADPLL de la Fig. 2.12).

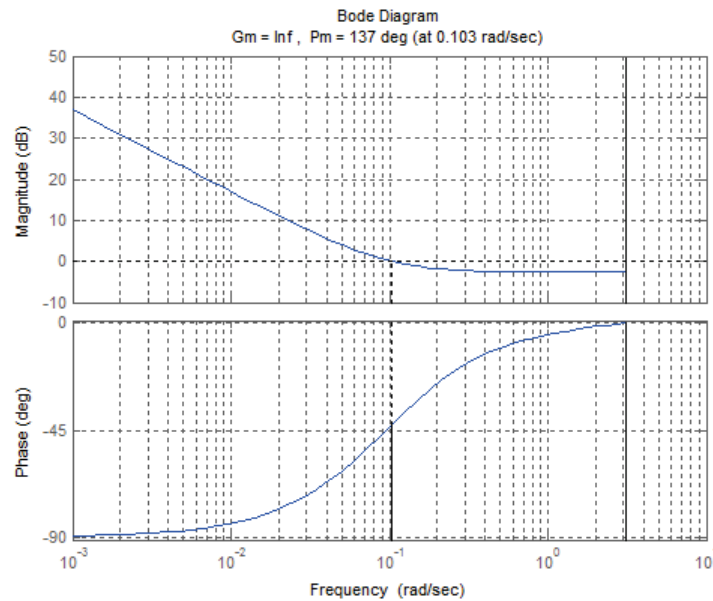


Fig. 2.21 – Diagramme de Bode d'un correcteur PI pour un couple de coefficient donné.

2.3.3. L'oscillateur à commande digitale (DCO)

Bien que l'étude théorique du DCO se base uniquement sur les équations de modélisation de celui-ci comme un simple intégrateur de phase discret, nous tenons quand même dans cette partie à présenter le DCO fonctionnel conçu pour notre application de distribution d'horloge. Il s'agit d'un DCO de 10 bits, réalisé en technologie

CMOS par E. Zianbetov et dont la fréquence peut varier de 1.1 à 2.8 GHz [62]. La conception de ce DCO a été validée par une simulation post-layout. La sortie de l'oscillateur a montré une bonne linéarité dans la gamme de fréquence fonctionnelle (ce qui encore une fois donne une bonne crédibilité à la modélisation linéaire du DCO) avec une moyenne de consommation de 6mW/GHz pour une tension d'alimentation de 1.1V. Pour ce qui concerne le bruit de phase du DCO, il a été mesuré à -86.12dBc/Hz pour une fréquence offset de 1MHz et une fréquence porteuse de 2 GHz. Il s'agit d'un oscillateur en anneau conçu à base d'inverseurs.

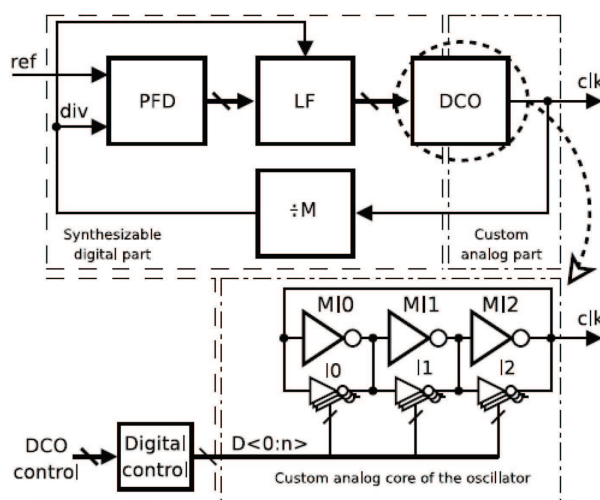


Fig. 2.22 – Implémentation du DCO d'une ADPLL.

L'architecture est basée sur un oscillateur en anneau de 7 étages avec une connexion en parallèle avec des inverseurs à 3 états dans chaque étage de l'oscillateur. Des détails plus précis en ce qui concerne la conception de ce DCO sont donnés en [62].

Paramètres	Valeur
Résolution	10 bits
Ajustement fin	1.6094 MHz/LSB
Ajustement grossier	6.4075 MHz/LSB
Gamme de Fréquence	1142 MHz – 2789 MHz
Bruit de phase ¹	-86.12 dBc/Hz @ 1 MHz
Consommation ²	15.678 mW @Fmax or ~ 6mW/GHz

Tableau 2.2 – Caractéristiques principales du DCO.

¹ Pour une fréquence centrale d'environ 2 GHz.

² Pour 1.1 V de tension d'alimentation.

La Fig. 2.23 montre l'évolution de la fréquence du DCO dans la plage fréquentielle de fonctionnement, on peut y constater une bonne linéarité. Le résultat de la simulation du bruit de phase est quant à lui donné en Fig. 2.24 pour une fréquence centrale de 2GHz correspondant à un code de contrôle égal à 201_{16} .

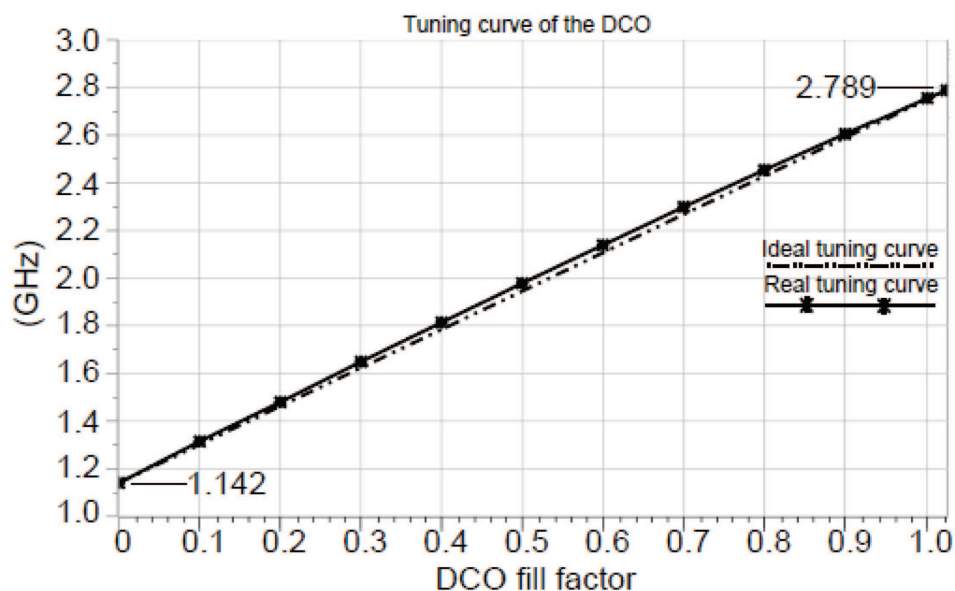


Fig. 2.23 – Évolution de la fréquence du DCO en fonction du code digital de contrôle.

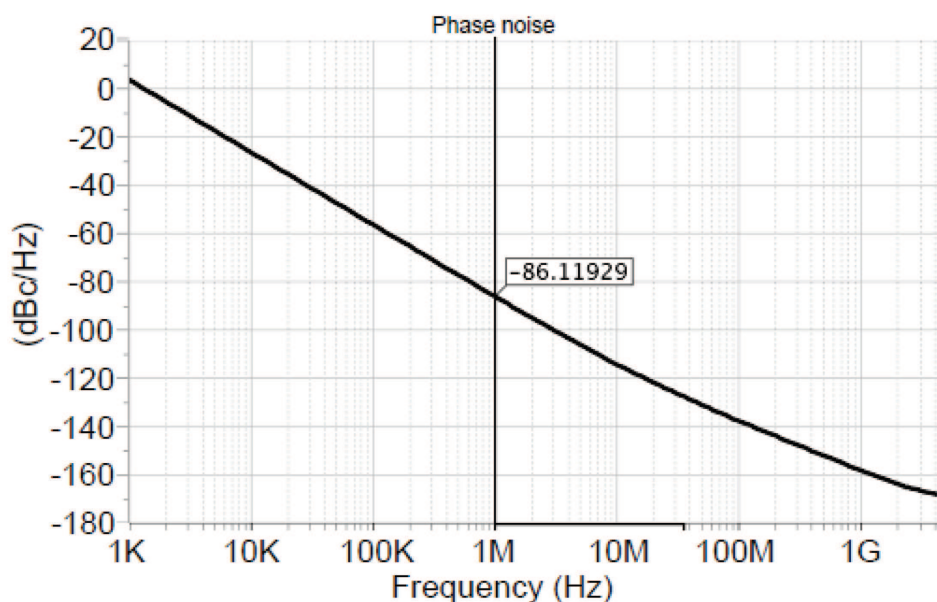


Fig. 2.24 – Bruit de phase pour une fréquence centrale de 2GHz.

2.3.4. Description d'un nœud sans auto-échantillonnage

Avant d'étudier les ADPLLs en prenant en considération des phénomènes non-linéaires, nous allons d'abord introduire le modèle théorique d'ADPLL simplifié, c'est-à-dire une ADPLL avec un échantillonnage uniforme. Ce modèle va servir comme point de référence afin d'évaluer comment évolue le domaine de stabilité de la SS-ADPLL qu'on va étudier par la suite en fonction de celui de l'ADPLL idéale.

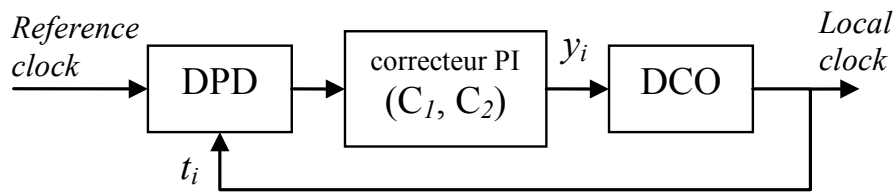


Fig. 2.25 – Modèle d'une ADPLL sans auto-échantillonnage.

On considère le modèle d'ADPLL sans auto-échantillonnage de la Fig. 2.25. La sortie $e_{ri}[n]$ du détecteur de phase est égale à la différence entre le temps d'arrivée du $n^{\text{ième}}$ front montant de l'horloge de référence et celle du $n^{\text{ième}}$ front montant de l'horloge générée localement, soit :

$$e_{ri}[n] = t_r[n] - t_i[n] \quad (2.9)$$

Le correcteur numérique $H(z)$ délivre au VCO un mot de commande $y[n]$ vérifiant la relation de récurrence suivante :

$$y[n] = y[n-1] + C_1 e_{ri}[n] + C_2 e_{ri}[n-1] \quad (2.10)$$

$$\text{avec } C_1 = K_p + K_i \text{ et } C_2 = -K_p$$

Le VCO modifie sa fréquence de fonctionnement en fonction de $y[n]$ selon :

$$t_i[n+1] = t_i[n] + T_i + g \cdot y[n] \quad (2.11)$$

Dans le cas où l'horloge de référence est une horloge de fréquence constante, dont les fronts montants sont donnés par :

$$t_r[n+1] = t_r[n] + T_r \quad (2.12)$$

on obtient aisément la relation vérifiée par l'erreur $e_{ri}[n]$ en soustrayant (2.11) à (2.12), ce qui donne :

$$e_{ri}[n+1] = e_{ri}[n] + (T_r - T_i) - g \cdot y[n] \quad (2.13)$$

ou à l'instant précédent :

$$e_{ri}[n] = e_{ri}[n-1] + (T_r - T_i) - g \cdot y[n-1] \quad (2.14)$$

En soustrayant (2.13) à (2.14), on obtient :

$$e_{ri}[n+1] - 2e_{ri}[n] + e_{ri}[n-1] = -g(y[n] + y[n-1]) \quad (2.15)$$

ce qui, d'après (2.10) peut également s'écrire :

$$e_{ri}[n+1] - 2e_{ri}[n] + e_{ri}[n-1] = -(K_1 e_{ri}[n] + K_2 e_{ri}[n-1]) \quad (2.16)$$

$$\text{avec } K_1 = g \cdot C_1 \text{ et } K_2 = g \cdot C_2$$

ou encore :

$$e_{ri}[n+1] - (2 - K_1)e_{ri}[n] + (1 + K_2)e_{ri}[n-1] = 0 \quad (2.17)$$

La stabilité de (2.17) s'étudie de manière triviale : quelles que soient les conditions initiales, on peut montrer que la convergence à 0 de $e_{ri}[n]$ (qui définit un état synchronisé) ne dépend que des paramètres K_1 et K_2 . De plus, il est évident que le système synchronise si et seulement si celui-ci est stable. La détermination du *domaine de stabilité* du système (les valeurs de K_1 et K_2 pour lesquelles le système est stable) permet donc également de savoir pour quelles valeurs de K_1 et K_2 le système synchronise, aux considérations de modularité près. Dans le cas présent, on trouve le domaine de stabilité représenté à la Fig. 2.26, correspondant aux valeurs de K_1 et K_2

pour lesquelles le système autonome défini par (2.17) a tous ses pôles à l'intérieur du cercle unité - ou, de manière équivalente, pour lesquelles le polynôme caractéristique de (2.17) a des racines de module inférieur à 1. Ce résultat est aussi validé par simulations transitoires. Notons qu'en ce qui concerne toutes les simulations transitoires réalisées tout au long de ce rapport, le critère de convergence est défini de sorte à obtenir une valeur finale de l'erreur temporelle plus petite que 10^{-5} .

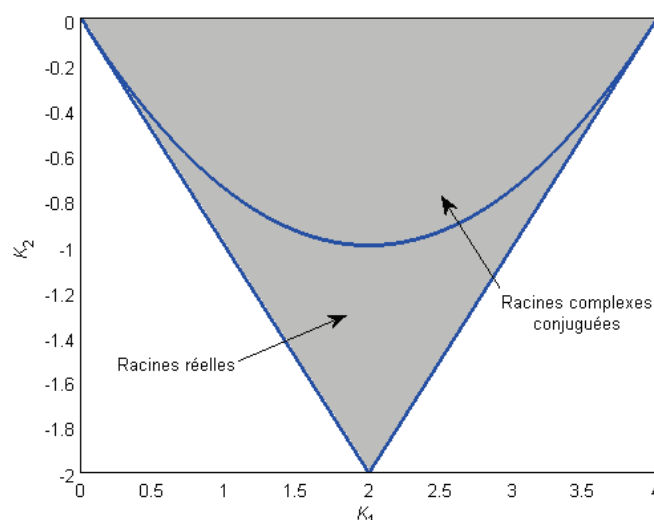


Fig. 2.26 – Domaine de stabilité d'une ADPLL classique sans auto-échantillonnage.

2.3.5. Modélisation de la SS-ADPLL et effet self-sampling

Une spécificité du réseau d'ADPLLs développé dans le cadre du projet HODISS est que les ADPLLs sont auto-échantillonnées. C'est-à-dire que les opérations du filtre numérique $H(z)$ sont cadencés par les fronts montants de l'horloge locale générée par le DCO. Ce choix se justifie par le fait de vouloir maximiser l'autonomie du système et ainsi éviter d'avoir recours à des horloges externes. En effet, il est apparu superflu de créer de façon externe une ou plusieurs horloges de fréquences pures, rien que pour faire fonctionner les filtres numériques. Cette démarche aurait été en opposition par principe avec l'objectif recherché.

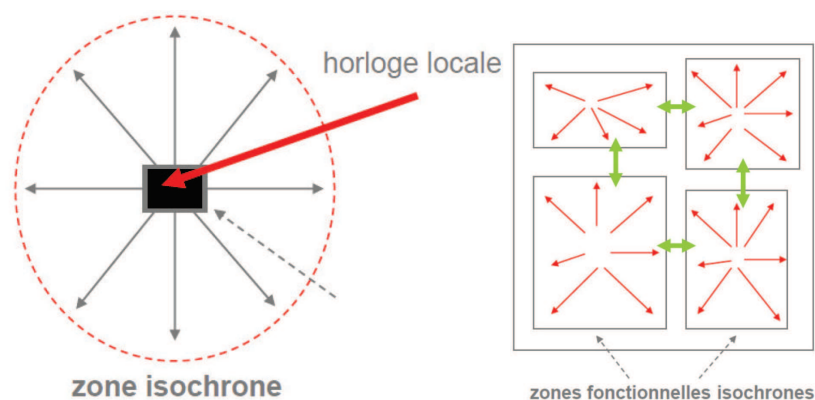


Fig. 2.27 – Zone isochrone avec en son centre son horloge locale (à gauche) et ensemble de zones isochrones échangeant entre elles des informations d'erreurs.

On fait donc en sorte que chaque zone isochrone soit au maximum de son autonomie avec uniquement des connections permettant d'acheminer en son entrée les informations en provenance des nœuds voisins.

Compte tenu de l'auto-échantillonnage non uniforme en régime transitoire, la fréquence de l'horloge locale en sortie du DCO variant dans le temps, il est légitime d'étudier l'influence cette spécificité sur le comportement d'une SS-ADPLL et par la suite sur des réseaux de SS-ADPLLs sachant l'importance qu'à le filtre numérique dans la dynamique du système. En effet, si l'on peut s'attendre à ce que la non uniformité de l'échantillonnage n'ait que peu d'influences sur le fonctionnement de la SS-ADPLL lorsque les paramètres de celle-ci (coefficients du filtre, gain du VCO) ont été choisis pour qu'elle réagisse lentement aux fluctuations de l'entrée (la non-uniformité est alors faible), il est impossible de prédire son comportement global sans une étude plus poussée. Il est donc important de modéliser une SS-ADPLL et de donner les équations régissant le système en tenant compte de cet aspect des choses. Cela revient à reconsidérer le fonctionnement du filtre PI classique donné en Fig. 2.20. En particulier, nous montrons comment différentes implémentations du filtre de boucle peuvent résulter en des dynamiques différentes du système. Nous allons ici n'en distinguer que trois types de fonctionnement : les SS-ADPLLs passéistes (type A et type B), les SS-ADPLLs nihilistes et les SS-ADPLLs avec prédiction auto régressive (AR) qui sont un corollaire des SS-ADPLLs passéistes et qui par conséquent s'étudient suivant la même méthode.

2.3.5.1. Les SS-ADPLLs passéistes

La Fig. 2.28 présente un schéma bloc détaillé du détecteur de phase et de fréquence et permet ainsi d'identifier tous les signaux en présence depuis l'entrée du détecteur DPD jusqu'à l'entrée du filtre numérique.

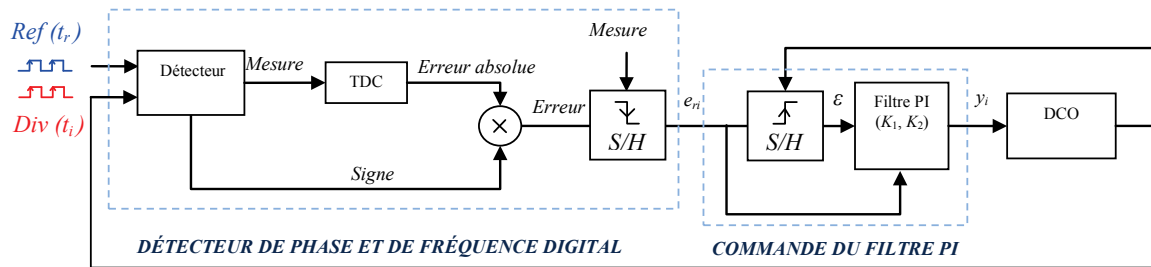


Fig. 2.28 – Diagramme bloc d'une SS-ADPLL passéiste.

Le chronogramme de la Fig. 2.29 présente l'évolution de ces signaux dans le temps pour un signal de référence supposé idéal et un signal d'horloge local choisi de sorte à pouvoir illustrer les variations possibles de celui-ci en fonction de la référence. La compréhension de ce chronogramme est importante dans la mesure où elle permet de comprendre toute la théorie développée pour expliquer le fonctionnement de la SS-ADPLL passéiste. Comme l'illustrent le signal '*Erreur*' et le signal '*e_{ri}*', le DPD envoie un signal qui est proportionnel à la différence de temps entre les instants où arrivent les fronts montants de l'horloge locale et ceux de l'horloge de référence. Lorsque l'horloge locale est en avance, la sortie du DPD est mise à jour après l'opération au niveau du filtre (Fig. 2.28 et Fig. 2.29), parce que le front montant de l'horloge de référence n'a pas encore été reçu. On voit arriver ce cas de figure sur le chronogramme de la Fig. 2.29, à partir des 7^e fronts montants des signaux d'entrée. En effet, comme le 7^e front montant de l'horloge locale arrive avant celui de la référence et que l'horloge locale est d'autre part utilisée comme référence pour piloter le filtre, celui-ci opère sans que le DPD n'ait eu le temps de rafraichir sa valeur, attendant l'arrivée du 7^e front montant de la référence. La valeur utilisée comme entrée par le filtre dans ce cas est la valeur disponible d'erreur à l'instant précédent comme sur le chronogramme (d'où la désignation de passéiste pour cette SS-ADPLL). Au 7^e front de l'horloge locale, on a $\varepsilon[7] = e_r[6]$.

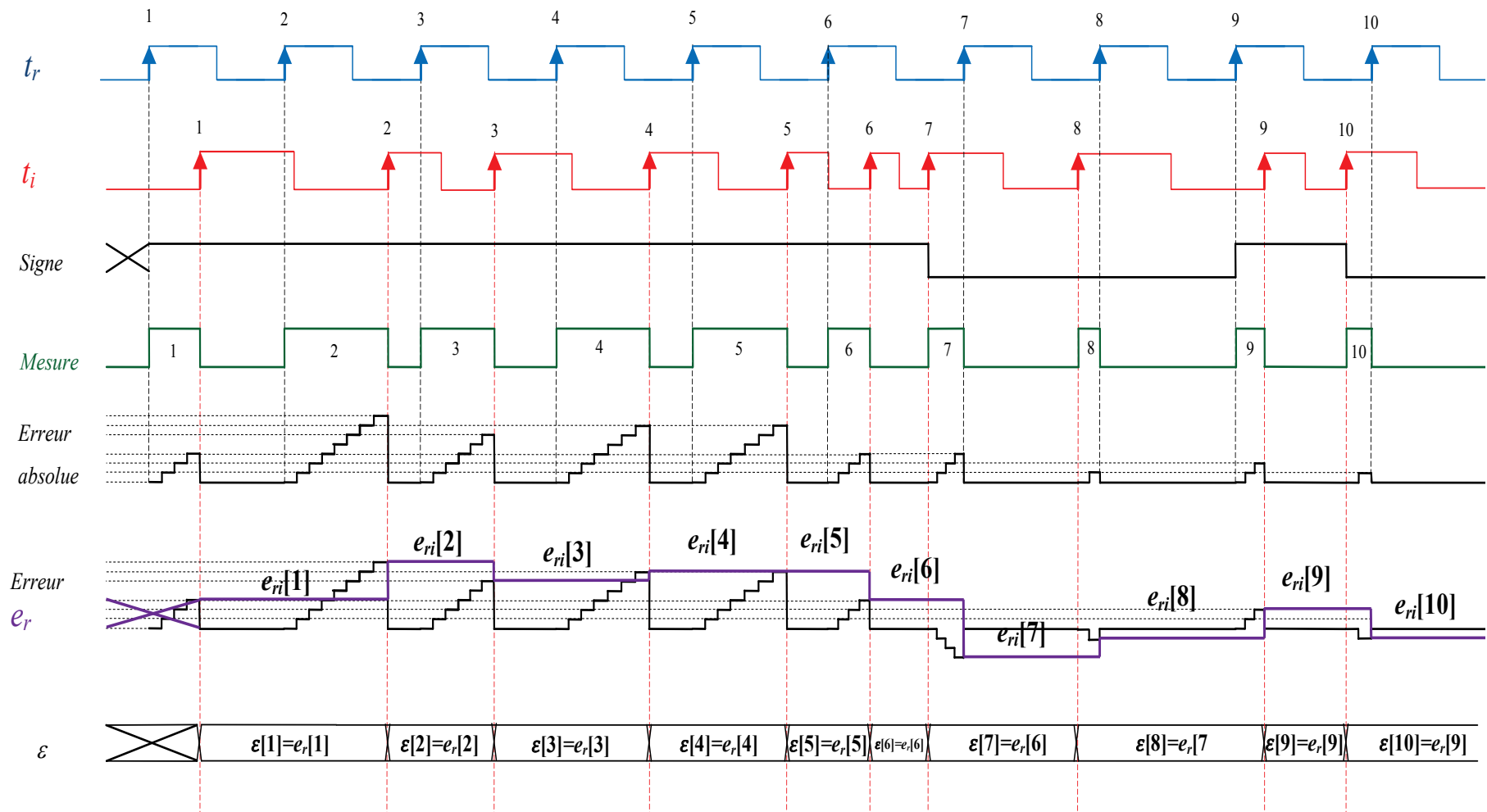


Fig. 2.29 – Chronogramme des signaux en présence de l'entrée de la SS-ADPLL jusqu'à l'entrée du filtre.

À cet égard, notre travail se distingue de nombreux autres travaux récents sur le sujet des ADPLLs, tels que [19-20,24], qui sont axés sur la non linéarité du DPD et négligent l'influence du séquençement des événements.

Les quantités $t_r[n]$ et $t_i[n]$ sont régies par les équations linéarisées:

$$t_r[n+1] = t_r[n] + T_r, \quad t_i[n+1] = t_i[n] + T_i + g \cdot y_i[n] \quad (2.18)$$

ou T_r est la période centrale du signal de référence, T_i celle de l'horloge locale, $y_i[n]$ le signal de commande en sortie du filtre PI rafraîchi à l'instant $t_i[n]$, et g un coefficient multiplicatif correspondant à un gain. Notons que dans le cas présent, c'est la période du DCO, qui est contrôlée, et non sa fréquence.

Maintenant, il est clair que si $t_i[n] < t_r[n]$ (c'est à dire que si l'horloge locale est en avance sur la référence), l'erreur temporelle $e_{ri}[n] = t_r[n] - t_i[n]$ ne peut pas être connue à l'instant $t_i[n]$. Une autre valeur est donc lue, valeur qui dans le cas présent des SS-ADPLLs passésistes correspond à :

$$\varepsilon[n] = e_{ri}[n-1] \quad (2.19)$$

Ce résultat est obtenu sachant que l'erreur $e_{ri}[n-1]$ est toujours connue à l'instant $t_i[n]$. En fonction de l'implémentation pratique du filtre, on peut distinguer deux types de SS-ADPLLs passésistes, celles de type A et celles de type B.

2.3.5.1.1. Les SS-ADPLLs passésistes de type A

Dans les SS-ADPLLs de type A, le filtre est implémenté de sorte à donner une quantité de contrôle $y_i[n]$ gouvernée par :

$$y_i[n] = y_i[n-1] + C_1 \varepsilon[n] + C_2 e_{ri}[n-1] \quad (2.20)$$

avec

$$\varepsilon[n] = \begin{cases} e_{ri}[n] & \text{if } e_{ri}[n] \leq 0 \\ e_{ri}[n-1] & \text{ailleurs} \end{cases} \quad (2.21)$$

C_1 et C_2 sont les coefficients du filtre. Cette configuration qui à certains points de vue parait la plus simple, du moins au niveau théorique suggère que l'erreur temporelle en entrée du filtre n'y est pas propagée. La partie intégrale est modulée uniquement par l'erreur temporelle à l'instant passé $e_{ri}[n-1]$. Les équations (2.20) et (2.21) correspondent à l'implémentation du filtre montré dans la Fig. 2.30. Rappelons qu'avec un fonctionnement idéal du filtre PI, c'est-à-dire sans auto-échantillonnage, le code de contrôle serait régi par l'équation (2.10):

$$y[n] = y[n-1] + C_1 e_{ri}[n] + C_2 e_{ri}[n-1]$$

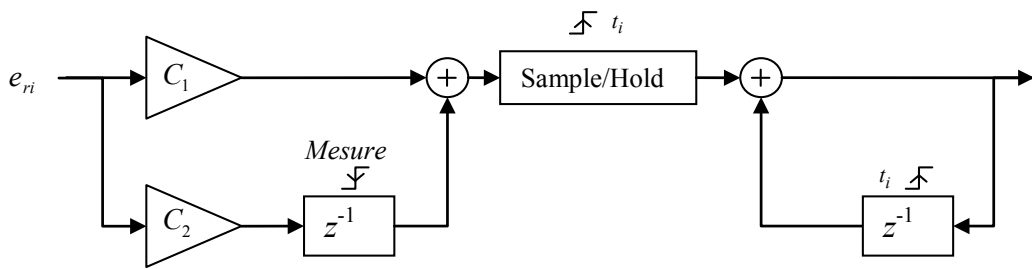


Fig. 2.30 – Mise en œuvre du filtre PI de type A.

De (2.18-2.20), on peut montrer que le système alterne en régime transitoire entre S_{retard} défini par l'équation (2.22) ci-dessous lorsque $e_{ri}[n] \leq 0$:

$$e_{ri}[n+1] - (2 - K_1)e_{ri}[n] + (1 + K_2)e_{ri}[n-1] = 0 \quad (2.22)$$

$$\text{avec } K_1 = g \cdot C_1 \text{ et } K_2 = g \cdot C_2$$

et S_{avance} défini par l'équation (2.23) ci-dessous lorsque $e_{ri}[n] > 0$:

$$e_{ri}[n+1] - 2e_{ri}[n] + (1 + K_1 + K_2)e_{ri}[n-1] = 0 \quad (2.23)$$

Les racines des polynômes caractéristiques associées aux équations de récurrence (2.22) et (2.23) sont respectivement :

$$\frac{2 - K_1 \pm \sqrt{K_1^2 - 4(K_1 + K_2)}}{2} \text{ et } 1 \pm \sqrt{-(K_1 + K_2)}.$$

Ainsi, la SS-ADPLL se comporte comme un système linéaire par morceaux de deuxième ordre commutant d'un état à l'autre. En fonction des valeurs de K_1 et de K_2 , ce système peut être globalement stable et converger vers un état synchronisé. Ceci va être étudié dans le point qui suit.

- **Analyse de la convergence et du comportement de la SS-ADPLL passiste de type A**

La convergence et le comportement de verrouillage globale de la SS-ADPLL passiste de type A ne dépend qu'en partie des racines des polynômes caractéristiques de (2.22) et (2.23). Premièrement, notons que S_{avance} est toujours instable. Si ses racines sont réelles, au moins l'une d'elles est positive et supérieure à 1 en module. Ainsi, selon les conditions initiales, la solution de (2.23) peut être croissante monotone, conduisant à une SS-ADPLL globalement instable. Une condition nécessaire pour la convergence globale est alors:

$$K_1 + K_2 > 0 \quad (2.24)$$

Si la condition (2.24) est satisfaite, S_{avance} a deux racines complexes conjuguées. En raison de la nature oscillatoire de la solution de (2.23), il existe toujours un entier naturel n qui fera en sorte que $e_{ri}[n] \leq 0$ et la SS-ADPLL commutera à S_{retard} . Deux cas devront désormais être pris en considération. Si S_{retard} a des racines réelles, la convergence est assurée si elles sont positives et inférieures à 1 (car alors la solution de (2.22) va décroître de façon exponentielle sans jamais revenir à l'état S_{avance}). Cela se traduit par les conditions suivantes:

$$\begin{cases} K_1^2 - 4(K_1 + K_2) > 0 \\ 1 + K_2 > 0 \\ 0 < K_1 < 2 \end{cases} \quad (2.25)$$

La première inégalité est la condition qui doit être satisfaite pour avoir des racines réelles. La condition sur le module de ces racines se traduit dans les deux autres inégalités. Les équations (2.24) et (2.25) définissent la zone grise rayée dans la Fig. 2.31. Il n'existe aucune méthode simple permettant de déterminer la convergence de la SS-ADPLL lorsque les racines de S_{retard} sont réelles, mais que l'une d'elle est négative. Enfin, si S_{retard} a des racines complexes conjuguées, à savoir si la condition :

$$K_1^2 - 4(K_1 + K_2) < 0 \quad (2.26)$$

est satisfaite, la SS-ADPLL reste constamment en commutation de S_{retard} à S_{avance} et vice versa. Les conditions pour lesquelles un tel système est globalement stable sont notoirement difficiles à déterminer. Au point suivant, nous allons introduire l'idée de la *moyennisation* du système en vue de son analyse de façon pragmatique.

- **Système moyen de la SS-ADPLL passéiste de type A**

Intuitivement, une condition nécessaire pour la convergence globale est que le système perde plus d'énergie lorsqu'il est régi par S_{retard} qu'il n'en gagne lorsqu'il est régi par S_{avance} . Deux approches peuvent être utilisées pour traduire cette intuition en termes mathématiques.

Tout d'abord, on peut ramener les équations (2.22) et (2.23) en une seule équation:

$$e_{ri}[n+1] - (2 - \frac{K_1}{2})e_{ri}[n] + (1 + \frac{K_1}{2} + K_2)e_{ri}[n-1] = \frac{K_1}{2} \text{sign}(e_{ri}[n])(e_{ri}[n] - e_{ri}[n-1]) \quad (2.27)$$

et on peut considérer la SS-ADPLL comme une perturbation d'un système "moyen" S_{moyen} régi par :

$$e_{ri}[n+1] - (2 - \frac{K_1}{2})e_{ri}[n] + (1 + \frac{K_1}{2} + K_2)e_{ri}[n-1] = 0 \quad (2.28)$$

Il serait alors raisonnable de supposer qu'une condition suffisante pour la convergence globale de la SS-ADPLL passéiste de type A est que S_{moyen} devrait être convergent. Il en résulte la condition suivante sur K_1 et K_2 :

$$K_1 + 2K_2 < 0 \quad (2.29)$$

Une autre approche qui peut être utilisée est de supposer que K_1 et K_2 sont petits. Les racines de S_{retard} et S_{avance} sont alors à proximité du cercle unité et il est rationnel de transformer les deux systèmes en temps continu (en utilisant l'invariance d'impulsion, par exemple). En supposant que l'état initial du système continu est:

$$e(0) = 0, \quad \dot{e}(0) = \dot{E}_0 > 0, \quad (2.30)$$

il est ensuite très simple de déterminer les instants t_n où se produisent les passages par zéro de $e(t)$. Les valeurs successives de $\dot{e}(t_{2n})$ définissent une suite géométrique: une condition suffisante pour la convergence du système transformé est que $\dot{e}(t_2) < \dot{E}_0$. Les calculs sont simples et, si nous restons dans la limite de petits K_1 et K_2 , la condition pour la convergence du système continu se ramène à (2.29).

Les simulations de la SS-ADPLL passéiste de type A sont données en Fig. 2.31. Pour chaque couple $\{K_1, K_2\}$, on note si la SS-ADPLL synchronise ou non. Les simulations montrent que, pour de petites valeurs de K_1 et K_2 , (2.29) est une condition de convergence, comme prévu. Pour des valeurs modérées de K_1 et K_2 , la frontière définie par (2.29) est respectée. Par ailleurs, on remarque qu'il existe des valeurs de K_1 et K_2 pour lesquelles S_{retard} a au moins un pôle réel négatif et pour lesquelles la synchronisation survient. La détermination rigoureuse du domaine de convergence de la SS-ADPLL passéiste de type A reste encore un problème ouvert.

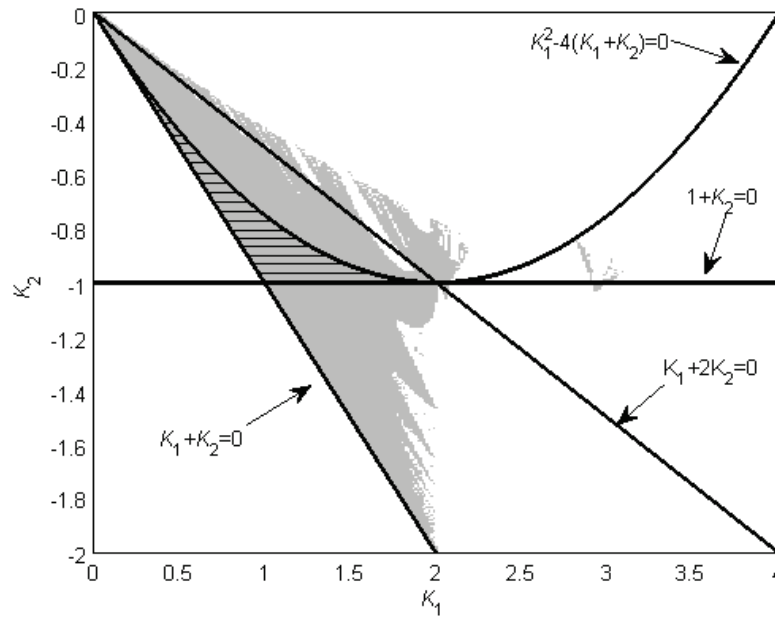


Fig. 2.31 - Domaine de stabilité d'une SS-ADPLL passéiste de type A. La zone grisée correspond aux valeurs de coefficients du filtre pour lesquelles les simulations ont montré que le SS-ADPLL synchronise.

2.3.5.1.2. Les SS-ADPLLs passéistes de type B

Dans les SS-ADPLLs passéistes de type B, le filtre est implémenté, comme dans la Fig. 2.32, de sorte à donner une quantité de contrôle $y_i[n]$ qui est gouvernée par :

$$y_i[n] = y_i[n-1] + C_1 \varepsilon[n] + C_2 \varepsilon[n-1] \quad (2.31)$$

avec

$$\varepsilon[n] = \begin{cases} e_{ri}[n] & \text{if } e_{ri}[n] \leq 0 \\ e_{ri}[n-1] & \text{ailleurs} \end{cases}$$

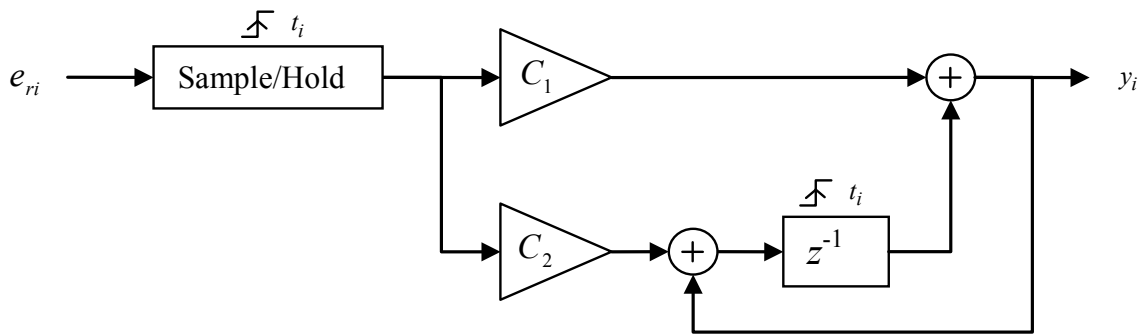


Fig. 2.32 – Mise en œuvre du filtre PI de type B.

C'est la configuration la plus naturelle d'un point de vue électronique issue d'un filtre PI (présenté en section 2.4.2). Elle suppose que la même erreur temporelle en entrée du filtre est propagée à travers celui-ci. De ce fait, aucune transformation du filtre PI classique n'est nécessaire. Par contre, comme on le verra par la suite, ce type de filtre rend le système plus complexe à étudier de façon théorique. De (2.18-2.19), (2.21) et (2.31), on peut montrer que l'erreur temporelle e_{ri} est régie par :

$$e_{ri}[n+1] = 2e_{ri}[n] - K_1\varepsilon[n] - e_{ri}[n-1] - K_2\varepsilon[n-1] \quad (2.32)$$

Notons que (2.32) peut être reformulée comme quatre équations linéaires séparées, où seulement $e_{ri}[n+1]$ et ses valeurs passées apparaissent, chaque équation correspondant à une valeur possible de $\varepsilon[n]$ et de $\varepsilon[n-1]$. La SS-ADPLL passéiste de type B synchronise quand $e_{ri}[n]$ tend vers zéro ou, de façon équivalente, lorsque le système linéaire par morceaux (PLS) défini par (2.32) est stable. L'étude de la stabilité d'une SS-ADPLL passéiste de type B est plus compliquée que celle de type A. Cela se traduit par la difficulté voire l'impossibilité de déterminer de manière analytique, des intervalles pour le choix des coefficients du filtre. Globalement, la détermination de la stabilité de systèmes SS-ADPLLs est une tâche notoirement difficile car il n'existe pas de méthode générique permettant de toutes les étudier de façon formelle. Même s'il est en revanche possible d'effectuer des simulations transitoires pour tous les types de SS-ADPLL, ces simulations ne peuvent pas non plus être entièrement fiables, car le comportement d'une SS-ADPLL peut dépendre de son état initial et tous les états initiaux possibles ne peuvent pas être testés. Afin de contourner cet inconvénient, il est possible, dans le cas des SS-ADPLLs de type B, d'utiliser l'approche introduite au chapitre 3 traitant de la stabilité des systèmes linéaires par morceaux dont les SS-ADPLLs sont un cas particulier.

- ***Système moyen de la SS-ADPLL passéiste de type B***

Cette approche, dont l'idée de départ a été donnée au deuxième point de la section 2.4.5.1.1, consiste à considérer une SS-ADPLL comme une perturbation d'un système moyen. **Plus précisément**, un système moyen est défini pour les SS-ADPLLs passéistes (resp. de type A et de type B) en remplaçant ε (resp. dans les équations (2.20) et (2.31)) par:

$$\tilde{\varepsilon}_{ri}[n] = \frac{1}{2}(e_{ri}[n] + e_{ri}[n-1]) \quad (2.33)$$

On trouve pour une SS-ADPLL pass  iste de type B, l'  quation gouvernant le syst  me moyen correspondant S_{moyen} donn  e par:

$$e_{ri}[n+1] - 2e_{ri}[n] + e_{ri}[n-1] = -\frac{K_1}{2}e_{ri}[n] - \left(\frac{K_1 + K_2}{2}\right)e_{ri}[n-1] - \frac{K_2}{2}e_{ri}[n-2]$$

soit S_{moyen} :

$$e_{ri}[n+1] - \left(2 - \frac{K_1}{2}\right)e_{ri}[n] + \left(1 + \frac{K_1}{2} + \frac{K_2}{2}\right)e_{ri}[n-1] + \frac{K_2}{2}e_{ri}[n-2] = 0 \quad (2.34)$$

Le syst  me r  sultant est d'ordre 3 et le principe de d  termination de la stabilit   est le m  me, c'est-  -dire le calcul du polyn  me caract  ristique, puis des racines de celui-ci. Les domaines de stabilit   obtenus par simulations transitoires du syst  me lin  aire par morceaux et analytiquement    l'aide du syst  me lin  aire moyen sont pr  sent  s en Fig. 2.33 pour les deux types de SS-ADPLLs pass  istes. M  me si cette approche conduit    une surestimation du domaine de stabilit  , elle reste quand m  me importante    consid  rer car comme on le verra par la suite dans l'  tude de la stabilit   des r  seaux cart  siens de SS-ADPLLs, elle donnera des r  sultats beaucoup plus satisfaisants en particulier pour les SS-ADPLLs pass  istes de type B.

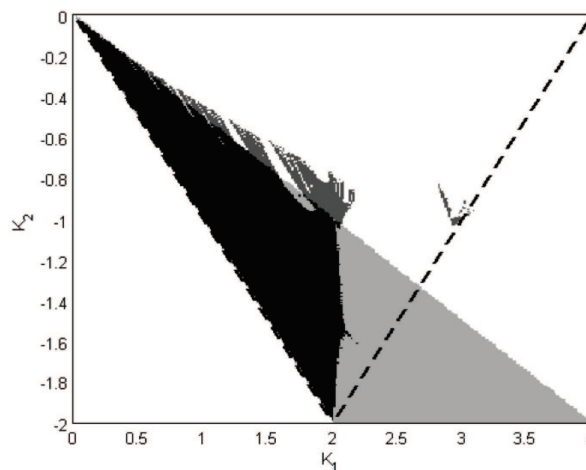


Fig. 2.33a. Domaine de stabilit   d'une SS-ADPLL pass  iste de type A. La zone grise claire a   t   obtenue en calculant les p  les du syst  me moyen, la zone grise fonc  e    celle obtenue par simulations transitoires et la zone noire est l'intersection des deux. Les lignes en pointill   repr  sentent les limites du domaine de stabilit   d'une ADPLL id  ale gouvern  e par l'  quation (2.17).

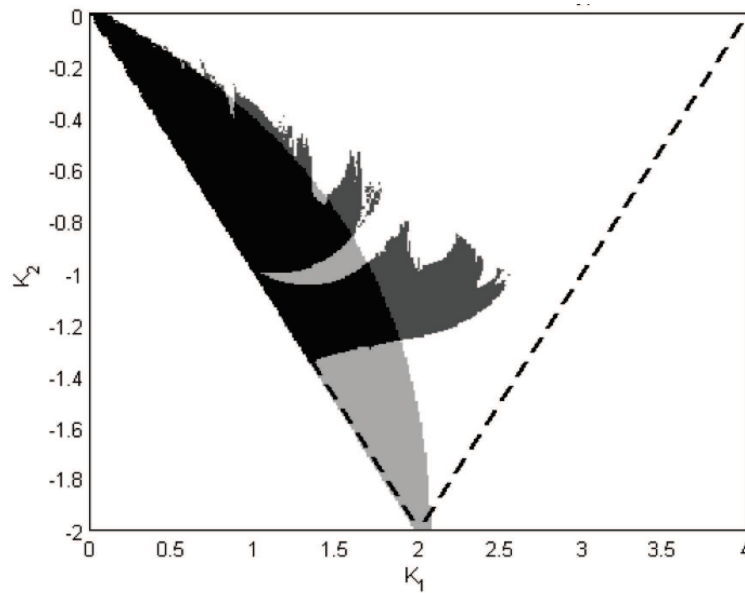


Fig. 2.33.b. Domaine de stabilité d'une SS-ADPLL passéiste de type B. La zone grise claire a été obtenue en calculant les pôles du système moyen, la zone grise foncée à celle obtenue par simulations transitoires et la zone noire est l'intersection des deux. Les lignes en pointillé représentent les limites du domaine de stabilité d'une ADPLL idéale gouvernée par l'équation (2.17).

2.3.5.2. La SS-ADPLL nihiliste

Il est possible d'envisager de programmer le filtre numérique de sorte à prendre une valeur nulle lorsque l'horloge locale est en avance par rapport à l'horloge de référence. Dans cette SS-ADPLL que nous avons nommé « nihiliste », on a :

$$\varepsilon[n] = \begin{cases} e_{ri}[n] & \text{si } e_{ri}[n] \leq 0 \\ 0 & \text{sinon} \end{cases} \quad (2.35)$$

En suivant la même démarche que dans la section 2.4.5.1 précédente, on montre que l'erreur temporelle $e_{ri}[n]$, pour une SS-ADPLL nihiliste de type A (code de contrôle $y_i[n]$ donné par l'équation (2.20)) est régie par les deux relations de récurrence :

$$S_{retard} : e_{ri}[n+1] - (2 - K_1)e_{ri}[n] + (1 + K_2)e_{ri}[n-1] = 0, \text{ si } e_{ri}[n] \leq 0, \quad (2.36)$$

et

$$S_{avance} : e_{ri}[n+1] - 2e_{ri}[n] + (1 + K_2)e_{ri}[n-1] = 0, \text{ sinon} \quad (2.37)$$

Il est possible de montrer simplement en prenant en compte des considérations analytiques, que la SS-ADPLL nihiliste de type A est instable, quelles que soient les valeurs de K_1 et K_2 . Cela se vérifie également par simulations transitoires.

La SS-ADPLL nihiliste de type B (code de contrôle $y_i[n]$ donné par l'équation (2.31)), est régie par l'équation (2.32). L'étude de la stabilité de cette SS-ADPLL n'étant pas possible par des méthodes analytiques, des simulations transitoires ont été effectuées et ont révélé que ce système est toujours instable. Même si ces résultats semblent a priori inintéressant, on verra par la suite que l'étude de la SS-ADPLL nihiliste peut être d'un certain intérêt pour les concepteurs car, au delà des simplifications que peuvent révéler sa conception tout comme ses équations d'état, sa mise en réseau permet de mettre en évidence des domaines de stabilité et de convergence.

- **Système moyen des SS-ADPLLs nihilistes**

Les équations (2.36) et (2.37), comme au deuxième point de la section 2.4.5.1.1, peuvent se combiner en une seule équation :

$$e_{ri}[n+1] - (2 - \frac{K_1}{2})e_{ri}[n] + (1 + K_2)e_{ri}[n-1] = \frac{K_1}{2} \text{sign}(e_{ri}[n])(e_{ri}[n]) \quad (2.38)$$

Le système moyen correspondant, pour une SS-ADPLL nihiliste de type A est régi par :

$$S_{\text{moyen}} : e_{ri}[n+1] - (2 - \frac{K_1}{2})e_{ri}[n] + (1 + K_2)e_{ri}[n-1] = 0 \quad (2.39)$$

En fait, le système moyen peut être défini pour les SS-ADPLLs nihilistes (resp. de type A et de type B) en remplaçant ε (resp. dans les équations (2.20) et (2.31)) par:

$$\tilde{\varepsilon}_{ri}[n] = \frac{1}{2} e_{ri}[n] \quad (2.40)$$

On trouve pour une SS-ADPLL nihiliste de type B :

$$S_{moyen} : e_{ri}[n+1] - (2 - \frac{K_1}{2})e_{ri}[n] + (1 + \frac{K_2}{2})e_{ri}[n-1] = 0 \quad (2.41)$$

Il existe des conditions pour lesquelles le système moyen de la SS-ADPLL nihiliste de type A (resp. de type B) est stable car il est possible de trouver des conditions sur K_1 et K_2 pour lesquelles l'équation caractéristique de (2.39) (resp. (2.41)) a ses deux racines à l'intérieur du cercle unité. Ces résultats n'étant pas compatibles avec les observations des simulations transitoires, on peut conclure, en ce qui concerne les SS-ADPLLs nihilistes, que l'approche du système moyen n'est pas applicable.

2.4. Conclusion

L'élément de base d'un réseau de distribution d'horloge basé sur des SS-ADPLLs a été décrit dans ce chapitre. Malgré son apparente simplicité, le comportement d'un tel système est assez complexe à analyser. Certaines questions, comme la détermination rigoureuse et formelle de l'ensemble du domaine de stabilité et de synchronisation du système restent ouvertes. Il a été montré que la SS-ADPLL se comporte comme un système linéaire par morceaux et des conditions suffisantes de stabilité et de convergence ont été établies par des méthodes analytiques pour la SS-ADPLL de type A et vérifiées par des simulations. En ce qui concerne les SS-ADPLLs de type B, du fait du nombre d'état du système et des différentes transitions possibles entre elles, il n'a pas été possible de déterminer analytiquement des conditions suffisantes de stabilité. Une investigation plus approfondie des questions de convergence, pour la SS-ADPLL de type B, en utilisant des fonctions de Lyapunov quadratiques par morceaux est l'objet du prochain chapitre.

Chapitre 3 :

Étude de la stabilité au sens de Lyapunov

3.1. Introduction

D'après la théorie de la stabilité de Lyapunov, la stabilité de systèmes dynamiques peut être déterminée à partir de fonctions scalaires appelées fonctions de Lyapunov. Cela peut être fait autant pour des systèmes linéaires que pour des systèmes non-linéaires, en temps continu comme en temps discret. Considérons un système à temps continu linéaire invariant :

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t), \quad \mathbf{x}(t_0) = \mathbf{x}_0 \quad (3.1)$$

où $\mathbf{x}(t) \in R^n$ est le vecteur d'état du système et $\mathbf{A} \in R^{n \times n}$ sa matrice d'évolution.

Le système linéaire (3.1) est stable si toutes les valeurs propres de la matrice \mathbf{A} sont strictement à l'intérieur de la moitié gauche du plan complexe et si celles sur l'axe imaginaire sont des racines distinctes du polynôme caractéristique de \mathbf{A} .

Théorème 3.1

L'origine (zéro) d'un système dynamique invariant dans le temps est un point d'équilibre stable (au sens de Lyapunov) s'il existe une fonction scalaire $V(\mathbf{x})$ continument dérivable telle que sur toute trajectoire du système, les conditions suivantes sont réunies:

$$\begin{aligned} V(\mathbf{x}) &> 0, \quad V(\mathbf{0}) = 0 \\ \dot{V}(\mathbf{x}) &= \frac{dV}{dt} = \frac{\partial V}{\partial \mathbf{x}} \frac{d\mathbf{x}}{dt} \leq 0 \end{aligned} \quad (3.2)$$

Pour un système linéaire (3.1), une fonction de Lyapunov quadratique peut être choisie :

$$V(\mathbf{x}) = \mathbf{x}^T \mathbf{P} \mathbf{x}, \quad \mathbf{P} = \mathbf{P}^T > 0, \quad (3.3)$$

ce qui donne en utilisant (3.1) :

$$\dot{V}(\mathbf{x}) = \mathbf{x}^T (\mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{A}) \mathbf{x}$$

Donc le système est asymptotiquement stable si la condition suivante est satisfaite :

$$\mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{A} < 0 \quad (3.4)$$

Le dual du théorème 3.1 peut être formulé pour la stabilité de systèmes à temps discret.

En effet, pour un système linéaire à temps discret,

$$\mathbf{x}(n+1) = \mathbf{A}\mathbf{x}(n), \quad \mathbf{x}(0) = \mathbf{x}_0, \quad (3.5)$$

une fonction de Lyapunov de forme quadratique doit satisfaire :

$$V(\mathbf{x}) = \mathbf{x}^T(n) \mathbf{P} \mathbf{x}(n) > 0$$

$$\Delta V(\mathbf{x}) = V(\mathbf{x}+1) - V(\mathbf{x}) \leq 0$$

La condition de stabilité est donc la suivante :

$$\Delta V(\mathbf{x}) \leq 0 \Leftrightarrow \mathbf{A}^T \mathbf{P} \mathbf{A} - \mathbf{P} \leq 0. \quad (3.6)$$

S'il n'y a pas de méthodes génériques pour trouver des fonctions de Lyapunov pour les systèmes non-linéaires, il est possible pour les systèmes linéaires par morceaux, de définir une procédure de détermination de la stabilité par résolution d'inéquations matricielles appelées LMIs (Linear Matrix Inequalities). Nous allons introduire les outils applicables à la détermination de la stabilité des SS-ADPLLs en nous basant sur cette dernière approche. En effet, Johansson et Rantzer [27] ont proposé une méthode reposant sur des LMIs pour déterminer la stabilité des systèmes dits « linéaires par morceaux », tels que nos SS-ADPLLs. La méthode proposée consiste en la recherche d'une fonction de Lyapunov quadratique par morceaux, qui peut être formulée comme un problème d'optimisation convexe et résolu à l'aide d'outils LMIs standard [50, 66, 67]. L'adaptation de cette technique pour des systèmes à temps discret (notre cas) fut proposée par Feng [26].

3.2. Stabilité des systèmes discrets linéaires par morceaux

Un système discret linéaire par morceaux peut être représenté en vue de son analyse par :

$$\mathbf{x}(n+1) = \mathbf{A}_i \mathbf{x}(n), \quad \mathbf{x} \in S_i \quad (3.7)$$

où $\{S_i\}_{i \in I} \subset R^n$ est une partition de l'espace des états en un nombre de cellules fermées polyédriques, I est l'ensemble des indices des cellules, $\mathbf{x}(n) \in R^n$ les variables d'état du système et \mathbf{A}_i la matrice du i^{ieme} modèle local du système.

Supposons que lorsque la variable d'état du système transite de la région S_i à la région S_j à l'instant n , la dynamique du système est régie par la dynamique du modèle local de S_i à ce moment. Enfin, définissons par Ω l'ensemble représentant toutes les transitions possibles d'une région à l'autre, tel que:

$$\Omega = \{i, j \mid \mathbf{x}(n) \in S_i, \mathbf{x}(n+1) \in S_j, j \neq i\}. \quad (3.8)$$

3.2.1. Stabilité quadratique

Dans certains cas, il est possible de prouver la stabilité de systèmes linéaires par morceaux en utilisant une fonction de Lyapunov quadratique commune (CQLF) pour toutes les partitions. Les calculs sont basés sur les conditions suffisantes suivantes :

Proposition 3.1

S'il existe une matrice $\mathbf{P} = \mathbf{P}^T > 0$, telle que

$$\mathbf{A}_i^T \mathbf{P} \mathbf{A}_i - \mathbf{P} < 0, i \in I \quad (3.9)$$

alors la stabilité asymptotique du point zéro est garantie. Toutes les trajectoires $\mathbf{x}(n) \in \cup_{i \in I} S_i$ tendent vers l'origine. De plus, on dit que la fonction $V(n) = \mathbf{x}^T \mathbf{P} \mathbf{x}$ est une fonction de Lyapunov pour le système. La proposition précédente n'est qu'une condition suffisante de stabilité, dans le sens où un système à temps discret linéaire par morceaux peut ne pas satisfaire (3.9) mais être quand même stable [27]. Par exemple, dans le cas des SS-ADPLLs, la résolution de (3.9) ne permet jamais de trouver une CQLF, ce qui justifie le fait de chercher des fonctions de Lyapunov quadratiques par morceaux.

3.2.2. Stabilité quadratique par morceaux

Quand il n'est pas possible de trouver une fonction de Lyapunov quadratique globale (commune pour toutes les partitions), il existe peu de méthodes efficaces permettant malgré tout de construire un autre type de fonction de Lyapunov quadratiques (non commune par exemple) pour contourner le problème. Une extension naturelle de l'utilisation des fonctions de Lyapunov quadratiques globales est de considérer des fonctions qui sont elles aussi quadratiques par morceaux. En effet, il n'est pas nécessaire d'exiger une matrice définie positive commune \mathbf{P} pour un système linéaire par morceaux puisque la dynamique décrite par \mathbf{A}_i est uniquement valable à l'intérieur de la cellule S_i [27]. Par conséquent, une fonction de Lyapunov définie par morceaux pourrait être suffisante pour garantir la stabilité d'un système linéaire par morceaux [27]. Les travaux de Feng [26], s'inspirant du travail de Johansson et Rantzer, se situent dans cette idée et permettent de construire une fonction de Lyapunov définie par morceaux qui vérifie $\Delta V(n) = V(n+1) - V(n) \leq 0$ pour toutes les trajectoires du système. Considérons les fonctions de Lyapunov quadratiques par morceaux dans leur forme la plus simple en prenant un système avec deux cellules S_1 et S_2 . S'il existe \mathbf{P}_1 définie positive tel que la fonction $V_1(n) = \mathbf{x}^T \mathbf{P}_1 \mathbf{x}$ vérifie $\mathbf{A}_1^T \mathbf{P}_1 \mathbf{A} - \mathbf{P}_1 < 0$ dans la cellule S_1 et décroît lorsque la variable d'état du système transite vers S_2 et qu'il est possible de faire la même construction dans S_2 , alors la stabilité au sens de Lyapunov du système (3.7) est assurée.

En fait, il est possible de relâcher quelque peu les conditions de stabilité de Lyapunov pour analyser la stabilité de tels systèmes [50]. Il est possible de construire des matrices \mathbf{M}_i telles que $\mathbf{x}^T \mathbf{M}_i \mathbf{x} \geq 0$ pour $\mathbf{x} \in S_i$ en vue d'obtenir des conditions de stabilité relaxées du type:

$$\mathbf{A}_i^T \mathbf{P}_i \mathbf{A}_i - \mathbf{P}_i + \mathbf{M}_i < 0, i \in I \quad (3.10)$$

Notons que (3.10) implique (3.9) à l'intérieur de la cellule S_i . Notons également que, comme les cellules sont polyédriques, il est aisé de construire pour chacune d'entre elles une matrice \mathbf{E}_i telle que $\mathbf{E}_i \mathbf{x} \geq 0$, $\mathbf{x} \in S_i$, ce qui signifie qu'aucune des composantes de

$\mathbf{E}_i \mathbf{x}$ n'est négative. Alors, pour toute matrice positive \mathbf{U}_i (i.e. dont aucun des éléments n'est négatif), on a :

$$\mathbf{x}^T \mathbf{E}_i^T \mathbf{U}_i \mathbf{E}_i \mathbf{x} \geq 0, \mathbf{x} \in S_i, i \in I$$

et on peut donc poser $\mathbf{M}_i = \mathbf{E}_i^T \mathbf{U}_i \mathbf{E}_i$.

Le résultat principal du travail de Feng se résume au théorème suivant :

Théorème 3.1

Considérons le système linéaire par morceaux (3.7). S'il existe des matrices symétriques \mathbf{P}_i , \mathbf{U}_i , \mathbf{W}_i et \mathbf{Q}_{ij} , $i, j \in I$, telles que \mathbf{U}_i , \mathbf{W}_i et \mathbf{Q}_{ij} sont positives et que les LMIs suivantes sont respectées :

$$0 < \mathbf{P}_i - \mathbf{E}_i^T \mathbf{U}_i \mathbf{E}_i, i \in I \quad (3.11)$$

$$\mathbf{A}_i^T \mathbf{P}_i \mathbf{A}_i - \mathbf{P}_i + \mathbf{E}_i^T \mathbf{W}_i \mathbf{E}_i < 0, i \in I \quad (3.12)$$

$$\mathbf{A}_i^T \mathbf{P}_j \mathbf{A}_i - \mathbf{P}_i + \mathbf{E}_i^T \mathbf{Q}_{ij} \mathbf{E}_j < 0, i, j \in \Omega, \quad (3.13)$$

alors l'origine du système linéaire par morceaux est asymptotiquement stable et $\mathbf{x}(n)$ tend vers l'origine de manière exponentielle pour chaque trajectoire dans l'espace d'état.

Si on arrive à construire des matrices vérifiant les LMIs (3.5-3.7), alors la fonction $V(n)$ définie par :

$$V(n) = \mathbf{x}^T \mathbf{P}_i \mathbf{x}, \mathbf{x} \in S_i, \quad (3.14)$$

est une fonction de Lyapunov pour le système. La LMI (3.11) garantit que la fonction est positive dans chacune des cellules. La LMI (3.12) garantit que sa dérivée discrète est négative (décroissance) à travers toutes les trajectoires du système dans chaque région. La LMI (3.13) garantit la décroissance (perte d'énergie) lorsque la variable d'état du système effectue des transitions d'une région à l'autre.

3.2.3. Application aux SS-ADPLLs

Dans ce qui suit, une comparaison des domaines de stabilité obtenus par simulation transitoire et ceux obtenus par détermination des fonctions de Lyapunov pour des systèmes linéaires par morceaux est faite pour un nœud de SS-ADPLL.

3.2.3.1. Stabilité d'un nœud de SS-ADPLL passéiste de type A

Pour une SS-ADPLL passéiste de type A, l'erreur temporelle $e_{ri}[n]$ est régie par les équations (2.22) et (2.23). L'équation du système se ramène donc à la forme canonique:

$$\mathbf{x}(n+1) = A_i \mathbf{x}(n) \quad i \in \{1,2\}, \quad \mathbf{x}[n] = (e_r[n] \quad e_r[n-1])^T$$

$$\text{avec } \begin{cases} \mathbf{A}_1 = \begin{bmatrix} 2 - K_1 & -(1 + K_2) \\ 1 & 0 \end{bmatrix} & \text{si } e_{ri}[n] \leq 0 \\ \mathbf{A}_2 = \begin{bmatrix} 2 & -(1 + K_1 + K_2) \\ 1 & 0 \end{bmatrix} & \text{si } e_{ri}[n] > 0 \end{cases}$$

Seulement, comme la matrice \mathbf{A}_2 correspondant au système S_{avance} est celle d'un système instable, la condition (3.12) ne peut pas être respectée. De ce fait, la méthode des PQLFs, telle qu'elle est énoncée précédemment pour les PLSs ne peut pas être appliquée à cette SS-ADPLL passéiste. Il est à noter qu'il n'est pas possible de trouver une PQLF pour un système qui reste pour plus d'un pas d'échantillonnage dans une cellule instable (i.e. une cellule S_i dont la matrice d'état \mathbf{A}_i a une valeur singulière supérieure à 1). Des travaux plus récents traitant de la stabilité de PLSs en recherchant des fonctions de Lyapunov *non monotones* [52] peuvent éventuellement être plus probants pour la SS-ADPLL de type A mais cette approche n'a pas été développée ici faute de temps.

3.2.3.2. Stabilité d'un nœud de SS-ADPLL passéiste de type B

Pour une SS-ADPLL passéiste de type B, l'erreur temporelle $e_{ri}[n]$ est régie par (2.32). Cette dernière équation du système peut se mettre sous la forme canonique (3.7) en posant :

$$\mathbf{x}[n] = (e_{ri}[n] \quad e_{ri}[n-1] \quad e_{ri}[n-2])^T$$

$$\mathbf{A}_1 = \begin{bmatrix} 2-K_1 & -1-K_2 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad \text{si } e_{ri}[n] \text{ et } e_{ri}[n-1] \leq 0, \quad (3.15)$$

$$\mathbf{A}_2 = \begin{bmatrix} 2 & -1-K_1-K_2 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad \text{si } e_{ri}[n] > 0 \text{ et } e_{ri}[n-1] \leq 0, \quad (3.16)$$

$$\mathbf{A}_3 = \begin{bmatrix} 2 & -(1+K_1) & -K_2 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad \text{si } e_{ri}[n] \text{ et } e_{ri}[n-1] > 0, \quad (3.17)$$

$$\mathbf{A}_4 = \begin{bmatrix} 2-K_1 & -1 & -K_2 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad \text{si } e_{ri}[n] \leq 0 \text{ et } e_{ri}[n-1] > 0. \quad (3.18)$$

Contrairement aux SS-ADPLLs de type A, les SS-ADPLLs de type B traitées ici peuvent être soumises à l'approche PQLF parce que, par construction, le système ne peut pas rester plus d'un pas d'échantillonnage dans les cellules instables S_2 et S_4 . Les diagrammes de transition présentés en Fig. 3.1 illustrent bien cette caractéristique pour les deux types de SS-ADPLLs passéistes.

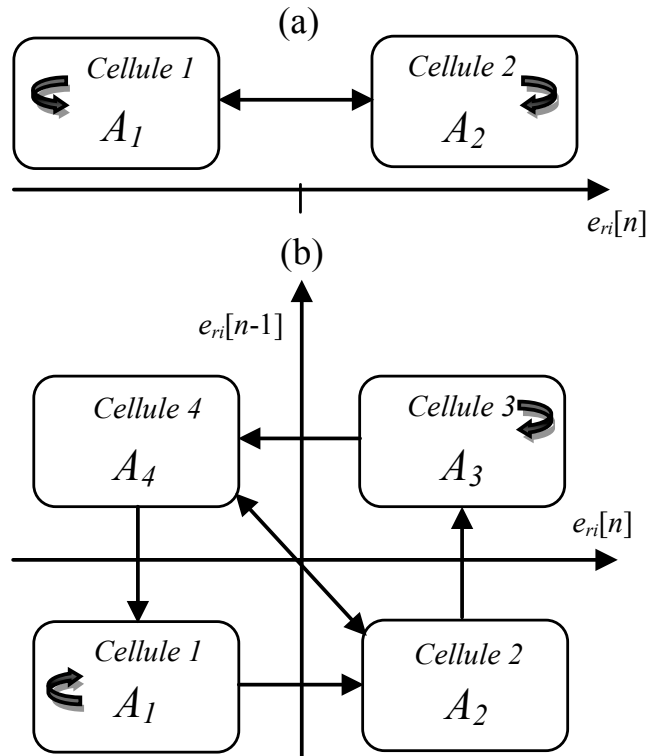


Fig. 3.1 - Diagramme de transition d'une SS-ADPLL passéiste de type A (a) et celui d'une SS-ADPLL passéiste de type B (b).

Pour la résolution des inégalités matricielles, on choisit:

$$\mathbf{E}_1 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \mathbf{E}_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \mathbf{E}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \text{ et } \mathbf{E}_4 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Les inégalités matricielles sont résolues sous MATLAB à l'aide d'un outil (Yalmip) permettant de trouver des solutions d'optimisation convexe du type LMI. Les résultats sont montrés à la Fig. 3.2 et les matrices \mathbf{P}_i trouvées sont définies positives.

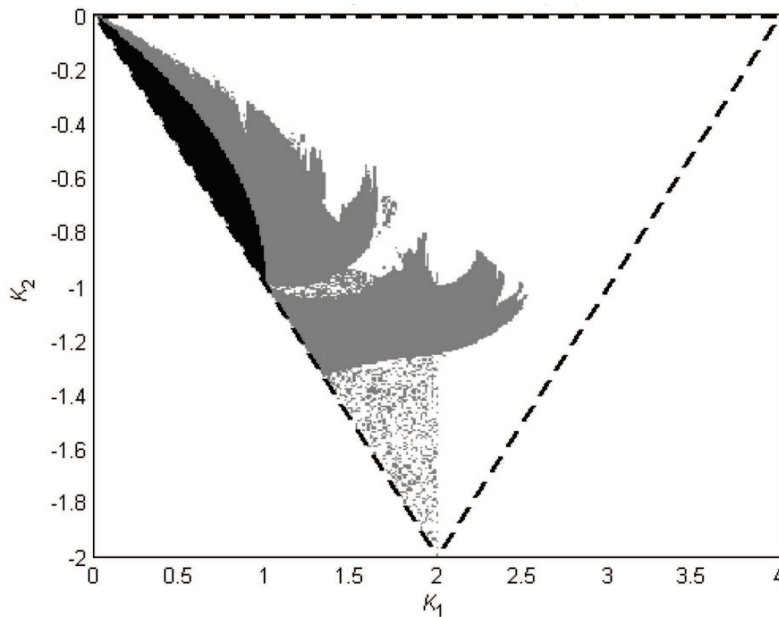


Fig. 3.2 – Domaine de stabilité d'une SS-ADPLL passiste de type B. La zone grisée est obtenue par simulation transitoire et la zone noire est obtenue par recherche de PQLFs en résolvant des LMIs.

Sur la Fig. 3.2, la zone noire correspond aux points de coordonnées (K_1, K_2) pour lesquelles le solveur a trouvé des matrices \mathbf{P}_i , \mathbf{U}_i , \mathbf{W}_i et \mathbf{Q}_{ij} vérifiant les LMIs (3.11), (3.12), (3.13). Il est important de rappeler ici que la méthode employée ne permet d'établir que des conditions suffisantes de stabilité et donc de convergence asymptotique pour des paramètres K_1 et K_2 donnés. La zone grise correspond aux valeurs de K_1 et K_2 pour lesquelles les simulations transitoires ont montré que le système converge après un nombre d'itérations inférieur à 10^5 . En conséquence, on ne doit pas s'étonner de voir que le domaine de stabilité prévu par la méthode de Feng soit plus petit que celui prévu par simulation transitoire. Les valeurs initiales de e_{ri} suivent une distribution normale $N(0,1)$. Dans la Fig. 3.2, le triangle délimité par les lignes en pointillé correspond au domaine de stabilité d'une SS-ADPLL "classique" pour laquelle $\varepsilon[n] = e_{ri}[n]$. Choisir K_1 et K_2 dans la zone grise devrait toujours se faire avec prudence. Comme on peut le voir sur la Fig. 3.2, la frontière entre la zone grise et la zone blanche (zone instable) est très difficile à définir avec précision. Par ailleurs, certains points appartenant à la zone grise peuvent éventuellement correspondre à des états instables du système. Par exemple, en choisissant $K_1 = 1.8$ et $K_2 = -1.4$ pour une SS-ADPLL

passéiste de type A et en testant des conditions initiales différentes de la forme $e_{ri}[1] = \sin \alpha$ et $e_{ri}[2] = \cos \alpha$, nous obtenons les bassins d'attraction de la Fig. 3.3.

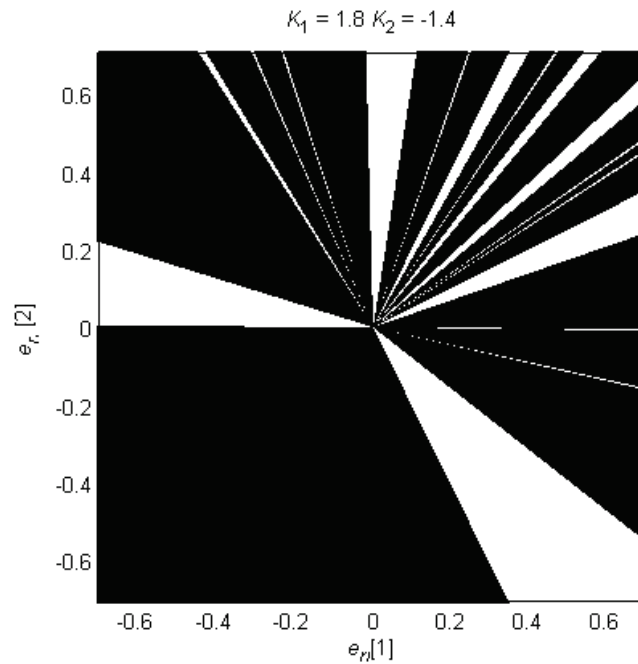


Fig.3.3 – Bassins d'attraction d'une SS-ADPLL passéiste de type A. La zone blanche (resp. noire) correspond aux conditions initiales conduisant à des trajectoires stables (resp. instables).

Selon les conditions initiales donc, cette SS-ADPLL peut synchroniser ou non. Cependant, pour de petites valeurs de $e_{ri}[n]$, le système peut facilement passer d'un comportement stable à un comportement instable (à cause d'erreurs d'arrondi par exemple).

3.3. Conclusion

Dans les sections précédentes, nous avons exploré quelques méthodes d'étude de la stabilité dont une basée sur des considérations analytiques et une autre sur des simulations transitoires. Cependant, pour cette dernière approche, il n'est possible d'établir la stabilité que pour un ensemble fini de conditions initiales. D'autre part, l'approche du système moyen ne s'est pas avérée totalement satisfaisante. L'approche développée dans ce chapitre, plus rigoureuse fut basée sur la construction d'une fonction de Lyapunov quadratique par morceaux (PQLF) [26-27]. Pour un PLS donné, on peut résoudre un système d'inégalités matricielles linéaires (LMI) pour déterminer si

une PQLF existe, dans ce cas, le système est stable. Cependant, il y a plusieurs problèmes avec cette approche : tout d'abord, l'existence d'une PQLF est seulement une condition suffisante pour la stabilité d'un système PLS. Par exemple, il y a plusieurs cas de SS-ADPLLs stables pour lesquelles aucune PQLF ne peut être construite, comme cela est montré dans [18]. Par ailleurs, la résolution d'un système de LMI devient très coûteuse en temps de calcul lorsque la taille du réseau augmente. Par exemple, si on considère un réseau de PLLs de taille N^2 , le nombre de matrices d'état du système sera de N^4 , d'où une démultiplication du nombre d'inconnues. Cette dernière considération est aussi valable pour les PLLs nihilistes. C'est la raison pour laquelle, dans le chapitre qui va suivre qui traite des réseaux de SS-ADPLLs, une autre approche sera proposée.

PARTIE II : LES RÉSEAUX DE BOUCLES A VERROUILLAGE DE PHASE DIGITALES

On s'intéresse dans cette seconde partie à la synchronisation de réseaux d'oscillateurs contrôlés de façon digitale et auto-échantillonnés. On va ne considérer que les *réseaux autonomes*, ce qui signifie qu'il n'existe pas d'horloge de référence absolue dans ses réseaux. Les résultats théoriques obtenus se basent sur cette configuration, le cas où il existe une référence absolue pouvant se traiter comme une extension du cas précédent. Seuls des résultats de simulations transitoires sont donnés pour les réseaux avec référence.

Cette seconde partie du rapport commence, au quatrième chapitre, par une étude consacrée aux réseaux de SS-ADPLLs de 2 nœuds. D'une façon intéressante, ces réseaux se sont avérés plus simples à étudier que les SS-ADPLLs prises individuellement et nous avons pu tirer des conditions nécessaires et suffisantes de stabilité. Pour des réseaux de plus grande taille, même si nous ne sommes pas arrivés à trouver d'une manière formelle des conditions suffisantes de stabilité, nous avons néanmoins réussi à obtenir une condition nécessaire de stabilité indépendante de leur taille. Dans le cas où le filtre de boucle est un filtre discret PI, paramétré par deux coefficients, nous montrons comment déterminer la région de l'espace des paramètres dans lequel les coefficients doivent être choisis en tant que pré requis pour la stabilité. Il se résume en de simples règles de conception, qui peuvent facilement être étendues à d'autres filtres. Cette condition nécessaire semble être suffisante dans le cas particulier des réseaux de SS-ADPLLs de type B. Nous vérifierons cette hypothèse par simulation en vue d'apporter une validation à cette propriété. De plus, l'approche de conception dite du « système moyen » que nous allons appliquer aux réseaux par la suite vient conforter cette hypothèse. Nous verrons donc que l'approche du système moyen est une bonne base pour les concepteurs en vue d'approcher et d'étudier les réseaux cartésiens de SS-ADPLLs de type B. Enfin, la validité de nos résultats théoriques sont illustrés par les résultats de simulation et au cinquième chapitre par des vérifications expérimentales sur le prototype FPGA [100] d'un réseau de 16 nœuds.

Chapitre 4 :

Synchronisation des réseaux cartésiens de SS-ADPLLs

4.1. Introduction

Ce chapitre est consacré à l'étude des réseaux cartésiens de SS-ADPLLs. Il est montré que les coefficients du filtre doivent satisfaire une certaine condition pour que le réseau entier soit stable, indépendamment de la taille du réseau. La validité des résultats théoriques développés dans cette partie est illustrée par des simulations. Le code Matlab utilisé pour les simulations transitoires de réseaux de différentes tailles est donné en annexes. Si la norme de $E_N_MINUS_1$ (vecteur regroupant les erreurs totales aux sorties de chaque nœud du réseau) est assez petit, après un nombre suffisant de fronts (selon la taille du réseau et les valeurs des coefficients K_1 et K_2), le réseau est supposé stable.

4.2. Stabilité des réseaux cartésiens autonomes de SS-ADPLLs

Nous traitons dans cette partie des réseaux cartésiens autonomes et auto-échantillonnés et déterminons les conditions sur les filtres de boucle permettant de synchroniser le réseau à la fois en phase et en fréquence. L'étude se focalise dans un premier temps sur les réseaux de 2 nœuds. Par la suite, les réseaux cartésiens de tailles quelconques sont étudiés.

4.2.1. Réseaux de 2 SS-ADPLLs

Bien que les résultats concernant la synchronisation d'un nœud de SS-ADPLL de type A et B avec une horloge de référence externe ne soient pas complètement concluantes, nous montrons ici que les petits réseaux autonomes de deux SS-ADPLLs identiques sont beaucoup plus simples à étudier. L'étude de la stabilité du réseau peut

être menée analytiquement. Par exemple, supposons que l'horloge de référence de la Fig. 2.28 soit remplacée par une autre SS-ADPLL du même type, comme indiqué dans la Fig. 4.1. On peut alors montrer que la synchronisation du réseau est gouvernée par une seule équation, dépendant uniquement du type de PLL utilisé.

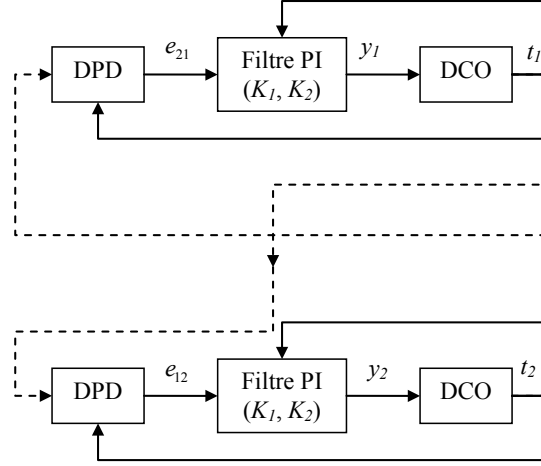


Fig. 4.1 - Réseau autonome de 2 SS-ADPLLs.

4.2.1.1. Réseaux de 2 SS-ADPLLs passésistes de type A

Sachant que $t_1[n+1] = t_1[n] + T_1 + g y_1[n]$,

on déduit que

$$t_1[n+1] - 2t_1[n] + t_1[n-1] = g(y_1[n] - y_1[n-1]) \quad (4.1)$$

qui se ramène à $t_1[n+1] - 2t_1[n] + t_1[n-1] = K_1 \varepsilon_1[n] + K_2 e_{21}[n-1]$.

De même, on a $t_2[n+1] - 2t_2[n] + t_2[n-1] = K_1 \varepsilon_2[n] + K_2 e_{12}[n-1]$

Donc

$$e_{12}[n+1] - 2e_{12}[n] + e_{12}[n-1] = K_1 (\varepsilon_1[n] - \varepsilon_2[n]) + K_2 (e_{21}[n-1] - e_{12}[n-1]) \quad (4.2)$$

On peut tirer parti du fait que

$$e_{21}[n] = -e_{12}[n] \quad (4.3)$$

pour simplifier l'équation d'une SS-ADPLL de type A en :

$$e_{12}[n+1] - 2e_{12}[n] + e_{12}[n-1] = K_1(\varepsilon_1[n] - \varepsilon_2[n]) - 2K_2e_{12}[n-1] \quad (4.4)$$

qui se ramène à :

$$e_{12}[n+1] - 2e_{12}[n] + e_{12}[n-1] = -K_1e_{12}[n] - (K_1 + 2K_2)e_{12}[n-1] \quad (4.5)$$

ou encore à :

$$e_{12}[n+1] - (2 - K_1)e_{12}[n] + (1 + K_1 + 2K_2)e_{12}[n-1] = 0 \quad (4.6)$$

Pour qu'un réseau de 2 SS-ADPLLs passésistes de type A synchronise, il est donc nécessaire et suffisant que (4.6) soit stable. La stabilité du réseau dépend en effet des racines du polynôme caractéristique de (4.6). Il est intéressant de noter qu'à travers l'équation (4.6), on est passé d'un problème non-linéaire à un problème linéaire.

4.2.1.2. Réseaux de 2 SS-ADPLLs passésistes de type B

En suivant le même raisonnement que pour la SS-ADPLL de type A, on trouve pour celle de type B :

$$e_{12}[n+1] - 2e_{12}[n] + e_{12}[n-1] = K_1(\varepsilon_1[n] - \varepsilon_2[n]) + K_2(\varepsilon_1[n-1] - \varepsilon_2[n-1]) \quad (4.7)$$

qui se ramène à :

$$e_{12}[n+1] - 2e_{12}[n] + e_{12}[n-1] = -K_1e_{12}[n] - (K_1 + K_2)e_{12}[n-1] - K_2e_{12}[n-2] \quad (4.8)$$

ou encore à :

$$e_{12}[n+1] - (2 - K_1)e_{12}[n] + (1 + K_1 + K_2)e_{12}[n-1] + K_2e_{12}[n-2] = 0 \quad (4.9)$$

4.2.1.3. Réseaux de 2 SS-ADPLLs nihilistes de type A

Pour la SS-ADPLL nihiliste de type A, on a comme en (4.3) :

$$e_{12}[n+1] - 2e_{12}[n] + e_{12}[n-1] = K_1(\varepsilon_1[n] - \varepsilon_2[n]) - 2K_2e_{12}[n-1].$$

donc

$$e_{12}[n+1] - 2e_{12}[n] + e_{12}[n-1] = -K_1 e_{12}[n] - 2K_2 e_{12}[n-1] \quad (4.10)$$

qui se ramène à :

$$e_{12}[n+1] - (2 - K_1)e_{12}[n] + (1 + 2K_2)e_{12}[n-1] = 0 \quad (4.11)$$

4.2.1.4. Réseaux de 2 SS-ADPLLs nihilistes de type B

Pour la SS-ADPLL nihiliste de type B, l'équation du système se ramène à :

$$e_{12}[n+1] - 2e_{12}[n] + e_{12}[n-1] = -K_1 e_{12}[n] - K_2 e_{12}[n-1] \quad (4.12)$$

ou encore à :

$$e_{12}[n+1] - (2 - K_1)e_{12}[n] + (1 + K_2)e_{12}[n-1] = 0 \quad (4.13)$$

Ces simplifications d'un système linéaire par morceaux à un système linéaire ne peuvent se faire que pour les géométries très particulières dont il est question ici. La stabilité des systèmes régis par (4.6), (4.9), (4.11) et (4.13) se fait avec les outils classiques de l'automatique.

Pour les 2 SS-ADPLLs passéistes de type A, on obtient les conditions de stabilité suivantes (Fig. 4.2) :

$$\begin{cases} K_1 + K_2 > 0 \\ 0 < K_1 < 4 \\ -2 < K_2 \\ K_1 + 2K_2 < 0 \end{cases} \quad (4.14)$$

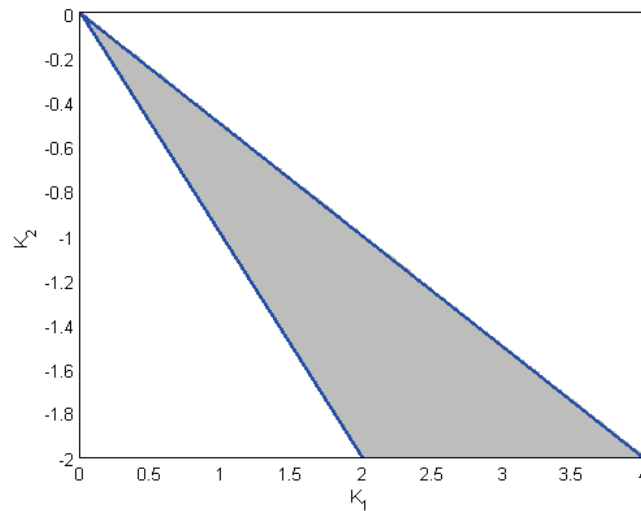


Fig. 4.2 - Domaine de stabilité (en grisé) d'un réseau de 2 SS-ADPLLs passésistes de type A.

Pour les 2 SS-ADPLLs passésistes de type B, on obtient analytiquement des conditions de stabilité dérivant des conditions pour lesquelles les modules des racines du polynôme caractéristique de (4.9) sont tous inférieurs à 1. Les racines sont calculées en appliquant la méthode de Cardan-Tartaglia. Ces conditions analytiques ne sont pas données à cause de leur expression trop longue. Cependant, le domaine correspondant à cette solution, confirmé par simulation transitoire est donné en Fig. 4.3.

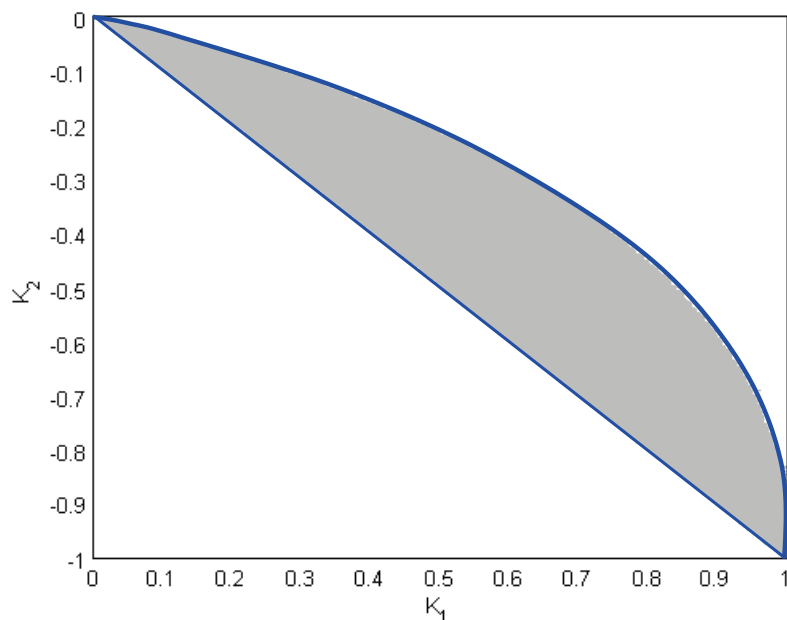


Fig. 4.3 - Domaine de stabilité (en grisé) d'un réseau de 2 SS-ADPLLs passésistes de type B.

Pour les 2 SS-ADPLLs nihilistes de type A, on obtient les conditions de stabilité suivantes (Fig. 4.4) :

$$\begin{cases} K_1 - 2K_2 < 4 \\ K_1 + 2K_2 > 0 \\ K_2 < 0 \end{cases} \quad (4.15)$$

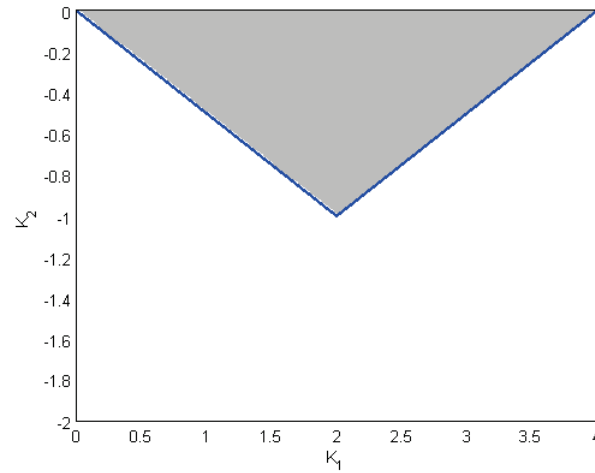


Fig. 4.4 - Domaine de stabilité (en grisé) d'un réseau de 2 SS-ADPLLs nihilistes de type A

Pour les 2 SS-ADPLLs nihilistes de type B, on obtient les conditions de stabilité suivantes (Fig. 4.5) :

$$\begin{cases} K_1 - K_2 < 4 \\ K_1 + K_2 > 0 \\ K_2 < 0 \end{cases} \quad (4.16)$$

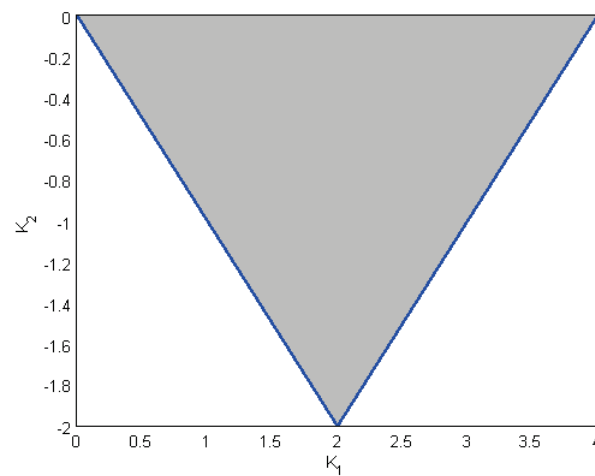


Fig. 4.5 - Domaine de stabilité (en grisé) d'un réseau de 2 SS-ADPLLs nihilistes de type B

Notons enfin que les zones de stabilité des Fig. 4.2, 4.3, 4.4 et 4.5 sont conservées pour des réseaux cartésiens 4 SS-ADPLLs. Par contre ces domaines de stabilité ne sont manifestement plus bons pour des réseaux de taille plus grande. On peut cependant montrer que, dans certains cas, il semble qu'on puisse ramener l'étude de la stabilité (de réseaux de taille supérieure à 4) à celle d'un système linéaire.

4.2.2. Réseaux passéistes de tailles quelconques

Cette partie analyse la stabilité de l'état synchronisé dans des réseaux cartésiens de SS-ADPLLs passéistes identiques de tailles quelconque pour des applications de distribution d'horloge. Sous les hypothèses de la section 3.2, ces réseaux peuvent être décrits comme des systèmes linéaires par morceaux, paramétrés par les coefficients du filtre de boucle, par le nombre de nœuds dans chaque ligne et chaque colonne du réseau. Les propriétés particulières de ces réseaux cartésiens de SS-ADPLLs vont permettre de définir un ensemble de règles de sélection des coefficients de chaque filtre de boucle qui doivent être respectés (pré requis) afin de réaliser la synchronisation dans n'importe quel réseau cartésien. Des simulations transitoires indiquent que cette condition nécessaire de synchronisation semble également être une condition suffisante pour les SS-ADPLLs passéistes de type B (celles développées dans le projet HODISS).. Le système considéré ici est composé de N nœuds identiques de SS-ADPLLs, dont chacun peut être considéré comme un oscillateur essayant de se caler sur la même phase que celle de ses voisins.

4.2.2.1. Définition du cadre

Un réseau cartésien typique est représenté à la Fig. 4.6. La sortie du DPD du k^{ieme} nœud au n^{ieme} front montant de l'horloge locale $t_k[n]$ est égale à:

$$\varepsilon_k[n] = \frac{1}{|V_k|} \sum_{l \in V_k} \varepsilon_{lk}[n], \quad (4.17)$$

avec

$$\varepsilon_{lk}[n] = \begin{cases} e_{lk}[n] & \text{si } e_{lk}[n] \leq 0 \\ e_{lk}[n-1] & \text{sinon} \end{cases}, \quad (4.18)$$

$$e_{lk}[n] = t_l[n] - t_k[n] = -e_{kl}[n], \quad (4.19)$$

où V_k est l'ensemble des indices des nœuds au voisinage du $k^{\text{ième}}$ nœud et $|V_k|$ est le cardinal de V_k , c'est à dire est égal à 2 pour les nœuds situés dans les coins, 3 pour les nœuds situés aux bords et 4 sinon. En utilisant (4.18) et (4.19), l'égalité fondamentale suivante peut être obtenue:

$$\varepsilon_{lk}[n] - \varepsilon_{kl}[n] = e_{lk}[n] + e_{lk}[n-1], \quad (4.20)$$

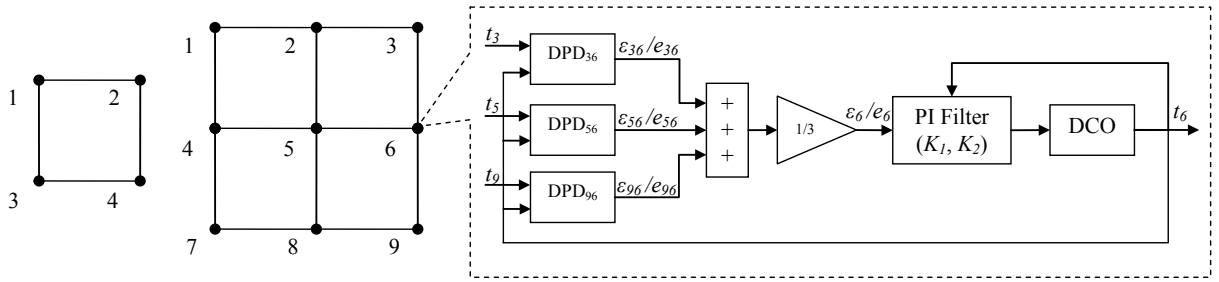


Fig. 4.6 - Deux réseaux cartésiens de 4 et 9 SS-ADPLLs. A chaque nœud, l'erreur totale est définie comme étant la valeur moyenne des sorties des DPDs.

Définissons un vecteur $\varepsilon[n]$ (resp. $\mathbf{t}[n]$ et $\mathbf{y}[n]$) dont la $k^{\text{ième}}$ coordonnée est $\varepsilon_k[n]$ (resp. $t_k[n]$ et $y_k[n]$) et $\mathbf{e}[n]$ un vecteur dont la $k^{\text{ième}}$ coordonnée est :

$$e_k[n] = \frac{1}{|V_k|} \sum_{l \in V_k} e_{lk}[n], \quad (4.21)$$

La $k^{\text{ième}}$ SS-ADPLL utilise $\varepsilon_k[n]$ (4.17) pour mettre à jour la sortie du filtre local a l'instant $t_k[n]$, comme dans la section 2.4.5. On peut alors assembler les équations qui régissent l'ensemble du réseau. De (4.19) et (4.21), on trouve :

$$\mathbf{e}[n] = -\mathbf{L}\mathbf{t}[n] \quad (4.22)$$

où \mathbf{L} est la matrice Laplacienne normalisée du réseau [91-92, 96, 99], définie comme suit:

$$L_{kl} = \begin{cases} 1 & \text{si } k = l \\ -\frac{1}{|V_k|} & \text{si } l \in V_k, \\ 0 & \text{sinon} \end{cases} \quad (4.23)$$

Pour tous les nœuds, indépendamment du type de filtre utilisé, nous avons :

$$\mathbf{t}[n+1] - 2\mathbf{t}[n] + \mathbf{t}[n-1] = g(\mathbf{y}[n] - \mathbf{y}[n-1]) \quad (4.24)$$

En multipliant (4.24) par \mathbf{L} et en utilisant (4.22), on trouve, pour les SS-ADPLLs de type A:

$$\mathbf{e}[n+1] - 2\mathbf{e}[n] + \mathbf{e}[n-1] = -\mathbf{L}(K_1 \boldsymbol{\varepsilon}[n] + K_2 \mathbf{e}[n-1]), \quad (4.25a)$$

et pour les SS-ADPLLs de type B

$$\mathbf{e}[n+1] - 2\mathbf{e}[n] + \mathbf{e}[n-1] = -\mathbf{L}(K_1 \boldsymbol{\varepsilon}[n] + K_2 \boldsymbol{\varepsilon}[n-1]) \quad (4.25b)$$

Étant donné un filtre de boucle arbitraire avec un pôle en zéro (afin d'assurer la synchronisation en phase), il sera toujours possible d'écrire les équations qui régissent le réseau sous la forme:

$$\mathbf{e}[n+1] - 2\mathbf{e}[n] + \mathbf{e}[n-1] = \mathbf{L} \left(\sum_{p=0}^P K_p \boldsymbol{\varepsilon}[n-p] + \sum_{q=0}^Q K'_q \mathbf{e}[n-q] \right), \quad (4.26)$$

Les équations (4.25a), (4.25b) et (4.26) sont des équations de systèmes linéaires par morceaux à cause des vecteurs $\boldsymbol{\varepsilon}[n]$. Il est cependant possible de tirer de chacune de ces dernières équations, une condition nécessaire de stabilité pour les réseaux cartésiens autonomes auto-échantillonnés.

Cela est possible en exploitant l'équation fondamentale (4.20) qui permet en effet d'obtenir à partir d'une différence d'éléments non linéaires, un résultat linéaire. Cette propriété est illustrée sur la Fig. 4.6a pour un réseau cartésien de 9 nœuds, où les erreurs $\varepsilon_{lk}[n]$ sont représentées par des flèches allant du nœud l vers le nœud k .

Sachant que l'erreur en sortie du DPD au k^{ieme} nœud est donnée par l'équation (4.17), les

flèches arrivant vers un nœud k représentent les erreurs qui seront sommées pour définir l'erreur totale $\varepsilon_k[n]$.

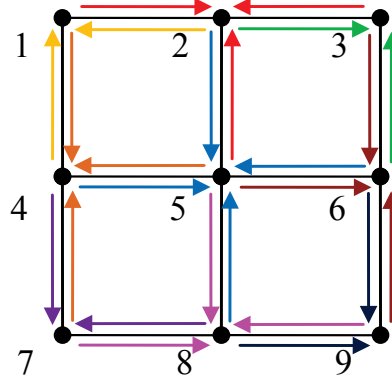


Fig. 4.6a - Propagation des erreurs dans chacune des branches dans un réseau cartésien autonome de 9 nœuds.

Une combinaison linéaire de ces erreurs totales pour tous les nœuds du réseau permet d'obtenir une équation linéaire. L'obtention de cette équation est toujours possible car sur chaque branche du réseau, il est possible d'appliquer l'équation (4.20). Pour le réseau de 9 nœuds de la Fig. 4.6a, l'équation peut se poser :

$$E[n] = 2\varepsilon_1[n] - 3\varepsilon_2[n] + 2\varepsilon_3[n] - 3\varepsilon_4[n] + 4\varepsilon_5[n] - 3\varepsilon_6[n] + 2\varepsilon_7[n] - 3\varepsilon_8[n] + 2\varepsilon_9[n]$$

Ce résultat se généralise pour un réseau de I lignes et de J colonnes. Un nœud numéro k étant donné, on peut définir sans ambiguïté l'indice des lignes i_k et l'indice des colonnes j_k telles que :

$$k = (i_k - 1)J + j_k, \quad (4.27)$$

où $i_k \in \{1, \dots, I\}$, $j_k \in \{1, \dots, J\}$.

Définissons un vecteur \mathbf{v} de taille IJ , tel que:

$$v_k = (-1)^{i_k + j_k} |V_k|, \quad (4.28)$$

Par exemple, pour le réseau 3 x 3 de la Fig. 4.6, nous avons:

$$\mathbf{v}^T = [2 \quad -3 \quad 2 \quad -3 \quad 4 \quad -3 \quad 2 \quad -3 \quad 2], \quad (4.29)$$

Enfin, introduisons «l'équation maîtresse » du réseau définie par la quantité :

$$E[n] = \mathbf{v}^T \mathbf{e}[n], \quad (4.30)$$

Définition 4.1 - *L'équation maitresse du réseau est l'équation scalaire obtenue par projection des équations qui régissent le réseau ((4.25a), (4.25b) et, plus généralement, (4.26)) sur \mathbf{v} (4.28).*

Cette équation maîtresse est une restriction du PLS (Piecewise Linear System) original à un sous-espace unidimensionnel, dans lequel le système reste confiné, si le vecteur d'état initial $\mathbf{e}[0]$ est colinéaire à \mathbf{v} . Par conséquent, il est évidemment nécessaire que l'équation maîtresse soit stable pour que l'ensemble du PLS soit lui aussi à son tour stable.

Maintenant, nous allons prouver que, pour un réseau de SS-ADPLLs, l'équation maîtresse est en fait linéaire. Il est alors facile de trouver une condition nécessaire pour la stabilité de l'ensemble du réseau, en calculant les racines de son polynôme caractéristique. Nous allons pour ce faire démontrer le théorème suivant:

Théorème 4.1 - *L'équation maîtresse d'un réseau cartésien autonome de SS-ADPLLs identiques est linéaire. Une condition nécessaire pour la stabilité et la synchronisation du réseau est alors que les racines du polynôme caractéristique de l'équation maîtresse doivent être à l'intérieur du cercle unité.*

Preuve du théorème 4.1

La preuve est organisée comme suit : Tout d'abord, nous allons prouver la propriété suivante :

Propriété 4.1 - *\mathbf{v}^T est le vecteur propre à gauche de la matrice Laplacienne normalisée associé à la valeur propre $\lambda = 2$.*

Ensuite, nous établissons :

Propriété 4.2 - Dans un réseau cartésien autonome de SS-ADPLLs identiques vérifiant (4.24), l'égalité suivante est vérifiée :

$$\mathbf{v}^T \boldsymbol{\varepsilon}[n] = \frac{1}{2} (E[n] + E[n-1]).$$

Si les propriétés 4.1 et 4.2 sont respectées, la projection de (4.25a), (4.25b) ou, plus généralement, (4.26) sur \mathbf{v} résulte alors en l'obtention d'une équation linéaire régissant $E[n]$, à partir de laquelle une condition nécessaire de stabilité peut être obtenue, prouvant ainsi le théorème 4.1.

a) Preuve de la Propriété 4.1

Soit $\mathbf{u}^T = \mathbf{v}^T \mathbf{L}$. Nous avons:

$$u_l = \sum_{k=1}^{IJ} v_k L_{kl} = \sum_{k=1}^{IJ} (-1)^{i_k+j_k} |V_k| L_{kl}. \quad (4.31)$$

De (4.23), le seul terme non nul sous le signe de la somme correspond aux valeurs $k=l$ ou $k/l \in V_k$. Maintenant nous constatons que, dans un réseau cartésien, si $l \in V_k$, nous sommes dans l'un des cas suivants:

$$i_l = i_k \pm 1$$

ou

$$j_l = j_k \pm 1.$$

Ainsi nous constatons que, si $l \in V_k$:

$$(-1)^{i_l+j_l} = -(-1)^{i_k+j_k}. \quad (4.32)$$

Ainsi (4.31) devient:

$$u_l = (-1)^{i_l+j_l} |V_l| - \sum_{\substack{k=1 \\ k/l \in V_k}}^{IJ} (-1)^{i_k+j_k} = (-1)^{i_l+j_l} \left(|V_l| + \sum_{\substack{k=1 \\ k/l \in V_k}}^{IJ} 1 \right). \quad (4.33)$$

Maintenant, le dernier terme du côté droit est le cardinal de V_l , de sorte que nous avons:

$$u_l = 2(-1)^{i_l + j_l} |V_l| = 2v_l. \quad (4.34)$$

Ceci termine la première étape de la preuve. Il est à noter que les valeurs propres de la matrice Laplacienne normalisée d'un réseau cartésien sont nécessairement inférieures à 2 [30]. Ainsi, \mathbf{v}^T est le vecteur propre associé à la valeur propre de plus grande magnitude du Laplacien normalisé.

b) Preuve de la Propriété 4.2

La deuxième étape de la preuve est simple, mais elle est assez fastidieuse. Premièrement, nous allons montrer que :

$$S = \sum_{k=1}^{IJ} (-1)^{i_k + j_k} \sum_{l \in V_k} \varepsilon_{lk} [n] + \varepsilon_{kl} [n] = 0. \quad (4.35)$$

Soit L_k (resp. C_k) l'ensemble des indices des nœuds appartenant à la même ligne (colonne resp.) que le nœud k . Il est clair que :

$$V_k = L_k \cup C_k. \quad (4.36)$$

Ainsi, S peut se réécrire :

$$S = \sum_{k=1}^{IJ} (-1)^{i_k + j_k} \sum_{l \in L_k} \varepsilon_{lk} [n] + \varepsilon_{kl} [n] + \sum_{k=1}^{IJ} (-1)^{i_k + j_k} \sum_{l \in C_k} \varepsilon_{lk} [n] + \varepsilon_{kl} [n]. \quad (4.37)$$

En dénommant S_L (resp. S_C) la somme sur les lignes (resp. colonnes), c'est à dire la première (resp. seconde) somme du second membre de l'équation (4.37), nous avons:

$$\begin{aligned}
S_L &= \sum_{k=1}^{IJ} (-1)^{i_k + j_k} \sum_{l \in L_k} \varepsilon_{lk}[n] + \varepsilon_{kl}[n] \\
&= \sum_{i=1}^I \sum_{j=1}^J (-1)^{i+j} \sum_{l \in L_{(i-1)J+j}} \varepsilon_{l,(i-1)J+j}[n] + \varepsilon_{(i-1)J+j,l}[n] \\
&= \sum_{i=1}^I (-1)^i \sum_{j=1}^J \sum_{l \in L_{(i-1)J+j}} (-1)^j (\varepsilon_{l,(i-1)J+j}[n] + \varepsilon_{(i-1)J+j,l}[n]) \\
&= \sum_{i=1}^I (-1)^i \Theta_i
\end{aligned} \tag{4.38}$$

On peut alors développer Θ_i :

$$\begin{aligned}
\Theta_i &= -(\varepsilon_{(i-1)J+2,(i-1)J+1}[n] + \varepsilon_{(i-1)J+1,(i-1)J+2}[n]) \Big\} j=1 \\
&+ (\varepsilon_{(i-1)J+2,(i-1)J+1}[n] + \varepsilon_{(i-1)J+1,(i-1)J+2}[n]) \Big\} j=2 \\
&+ (\varepsilon_{(i-1)J+3,(i-1)J+2}[n] + \varepsilon_{(i-1)J+2,(i-1)J+3}[n]) \Big\} j=3 \\
&- (\varepsilon_{(i-1)J+3,(i-1)J+2}[n] + \varepsilon_{(i-1)J+2,(i-1)J+3}[n]) \Big\} j=3 \\
&- (\varepsilon_{(i-1)J+4,(i-1)J+3}[n] + \varepsilon_{(i-1)J+3,(i-1)J+4}[n]) \Big\} j=4 \\
&+ \dots \\
&+ (-1)^{J-1} (\varepsilon_{(i-1)J+J-2,(i-1)J+J-1}[n] + \varepsilon_{(i-1)J+J-1,(i-1)J+J-2}[n]) \Big\} j=J-1 \\
&+ (-1)^{J-1} (\varepsilon_{(i-1)J+J,(i-1)J+J-1}[n] + \varepsilon_{(i-1)J+J-1,(i-1)J+J}[n]) \Big\} j=J-1 \\
&+ (-1)^J (\varepsilon_{(i-1)J+J,(i-1)J+J-1}[n] + \varepsilon_{(i-1)J+J-1,(i-1)J+J}[n]) \Big\} j=J
\end{aligned} \tag{4.39}$$

et constater que chaque ligne de (4.39) est annulé par la ligne suivante. Par conséquent, $\Theta_i = 0$ et $S_L = 0$. De même, on peut montrer que $S_C = 0$, ce qui prouve (4.35).

Nous pouvons compléter le reste de la démonstration preuve en écrivant :

$$\begin{aligned}
\mathbf{v}^T \mathbf{\varepsilon}[n] &= \sum_{k=1}^{IJ} (-1)^{i_k + j_k} |V_k| \varepsilon_k[n] \\
&= \sum_{k=1}^{IJ} (-1)^{i_k + j_k} \sum_{l \in V_k} \varepsilon_{lk}[n] - \varepsilon_{kl}[n].
\end{aligned} \tag{4.40}$$

A l'aide de (4.35), nous pouvons réécrire (4.40) :

$$\mathbf{v}^T \mathbf{\varepsilon}[n] = \frac{1}{2} \sum_{k=1}^{IJ} (-1)^{i_k + j_k} \sum_{l \in V_k} \varepsilon_{lk}[n] - \varepsilon_{kl}[n]. \tag{4.41}$$

qui se simplifie, en utilisant (4.20), en :

$$\begin{aligned}
\mathbf{v}^T \boldsymbol{\varepsilon}[n] &= \frac{1}{2} \sum_{k=1}^J (-1)^{i_k + j_k} \sum_{l \in V_k} e_{lk}[n] + e_{lk}[n-1] \\
&= \frac{1}{2} \sum_{k=1}^J (-1)^{i_k + j_k} |V_k| \left(\frac{1}{|V_k|} \sum_{l \in V_k} e_{lk}[n] + e_{lk}[n-1] \right). \\
&= \frac{1}{2} (E[n] + E[n-1])
\end{aligned} \tag{4.42}$$

Ceci achève la seconde partie de la preuve.

4.2.2.2. Discussion

Avant d'illustrer ces résultats, nous insistons sur le fait que les équations maîtresses obtenues par projection des équations de réseau sur \mathbf{v} sont indépendantes du nombre de lignes et de colonnes dans le réseau. Elles dépendent uniquement de la fonction de transfert du filtre de boucle et de son implémentation pratique.

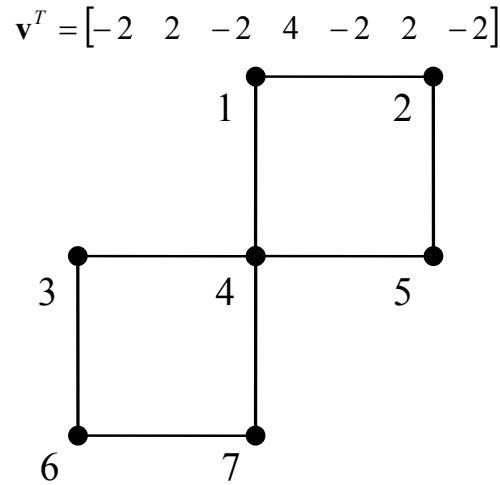


Fig. 4.7. Un réseau cartésien incomplet de 7 SS-ADPLLs avec le vecteur \mathbf{v}^T correspondant.

Par ailleurs, il est assez simple d'étendre le théorème 4.1 à des réseaux cartésiens incomplets, à savoir des réseaux cartésiens avec un ou plusieurs nœuds manquants (Fig. 4.7) et d'établir ainsi le corollaire suivant:

Corollaire 4.1 – *Le théorème 4.1 est valable pour n'importe quel réseau cartésien autonome et incomplet de SS-ADPLLs identiques.*

Les propriétés 4.1 et 4.2 restent inchangés même pour des réseaux incomplets, la seule différence réside dans la preuve de la propriété 4.2, où l'on doit faire attention à la présence de «trous» dans Θ_i .

Finalement, supposons que nous ayons choisi de prédire l'erreur temporelle d'une façon plus élaborée, de sorte à avoir :

$$\varepsilon_{lk}[n] = \begin{cases} e_{lk}[n] & \text{si } e_{lk}[n] \leq 0 \\ \sum_{i=1}^I a_i e_{lk}[n-i] & \text{ailleurs} \end{cases} \quad (4.43)$$

L'égalité fondamentale (4.20) devient alors:

$$\varepsilon_{lk}[n] - \varepsilon_{kl}[n] = e_{lk}[n] + \sum_{i=1}^I a_i e_{lk}[n-i]. \quad (4.44)$$

Il est alors possible de dériver une équation maîtresse en utilisant la propriété 4.1 et la propriété suivante:

Propriété 4.3 - *Dans un réseau cartésien autonome de SS-ADPLLs identique vérifiant (4.44), l'égalité suivante est vérifiée :*

$$\mathbf{v}^T \boldsymbol{\varepsilon}[n] = \frac{1}{2} \left(E[n] + \sum_{i=1}^I a_i E[n-i] \right).$$

La preuve de la propriété 4.3 est pratiquement la même que celle de la propriété 4.2, la seule différence est que c'est l'équation (4.44) (par opposition à (4.20)) qui est injectée dans (4.41), la partie antérieure de la preuve étant indépendante du fait que l'égalité fondamentale soit vérifiée. Ceci permet notamment d'appliquer les résultats de cette partie aux PLLs nihilistes.

4.2.3. Illustrations et résultats

4.2.3.1. Équations maîtresses des SS-ADPLLs passésistes de type A et B

Entreprenons, à titre d'illustration, de calculer les équations maîtresses des SS-ADPLLs de type A et de type B. La projection de (4.25a) sur \mathbf{v} conduit à :

$$E[n+1] - 2E[n] + E[n-1] = -K_1 \mathbf{v}^T \mathbf{L} \boldsymbol{\varepsilon}[n] - K_2 \mathbf{v}^T \mathbf{L} \mathbf{e}[n-1]. \quad (4.45)$$

En utilisant la propriété 4.1, (4.45) devient :

$$E[n+1] - 2E[n] + E[n-1] = -2K_1 \mathbf{v}^T \boldsymbol{\varepsilon}[n] - 2K_2 E[n-1]. \quad (4.46)$$

En utilisant la propriété 4.2, l'équation maîtresse de la SS-ADPLL de type A peut s'obtenir à partir de (4.46):

$$E[n+1] - 2E[n] + E[n-1] = -K_1 E[n] - (K_1 + 2K_2) E[n-1]. \quad (4.47a)$$

De même, pour la SS-ADPL de type B, on obtient:

$$E[n+1] - 2E[n] + E[n-1] = -K_1 E[n] - (K_1 + K_2) E[n-1] - K_2 E[n-2]. \quad (4.47b)$$

On constatera que le domaine de stabilité de l'équation maîtresse des réseaux de SS-ADPLLs de type A (resp. de type B) est le même que celui donné en Fig. 4.2 (resp. 4.3).

Il est intéressant de noter qu'il est toujours possible de trouver un réseau idéal de ADPLLs, c'est à dire qui ne soit pas auto-échantillonné, avec la même topologie que celui d'origine, dont l'équation maîtresse est la même que celle du réseau auto-échantillonné. Par exemple, (4.47a) et (4.47b) sont aussi les équations maîtresses (c'est-à-dire les projections sur \mathbf{v}) des réseaux régis par:

$$\mathbf{e}[n+1] - 2\mathbf{e}[n] + \mathbf{e}[n-1] = -\mathbf{L} \left(\frac{K_1}{2} \mathbf{e}[n] + \left(K_2 + \frac{K_1}{2} \right) \mathbf{e}[n-1] \right) \quad (4.48a)$$

et

$$\mathbf{e}[n+1] - 2\mathbf{e}[n] + \mathbf{e}[n-1] = -\mathbf{L} \left(\frac{K_1}{2} \mathbf{e}[n] + \left(\frac{K_1 + K_2}{2} \right) \mathbf{e}[n-1] + \frac{K_2}{2} \mathbf{e}[n-2] \right) \quad (4.48b)$$

Il est également intéressant de remarquer que les réseaux résultants idéaux sont en fait les «réseaux moyens», tels que définis dans les sections 2.4.5.1.1, 2.4.5.1.2 et 2.4.5.2, qui peuvent être obtenus à partir de (4.25a) ou (4.25b) par substitution de $\mathbf{e}[n]$ dans ces équations par :

$$\tilde{\mathbf{e}}[n] = \frac{1}{2} (\mathbf{e}[n] + \mathbf{e}[n-1]) \quad (4.49)$$

4.2.3.2. Équations maîtresses générales

En considérant la forme générale (4.26) et la prédiction de l'erreur temporelle (4.43), l'équation maîtresse se trouve être :

$$E[n+1] - 2E[n] + E[n-1] = \sum_{p=0}^P K_p E[n-p] + 2 \sum_{q=0}^Q K'_q E[n-q] + \sum_{p=0}^P \sum_{i=1}^I a_i K_p E[n-p-i] \quad (4.50)$$

Tout comme dans la section 4.2.3.2, le «réseau moyen» correspondant peut être obtenu en substituant $\mathbf{e}[n]$ dans (4.26) par :

$$\tilde{\mathbf{e}}[n] = \frac{1}{2} \left(\mathbf{e}[n] + \sum_{i=1}^I a_i \mathbf{e}[n-i] \right) \quad (4.51)$$

4.2.3.3. Domaines de stabilité de réseaux de SS-ADPLLs passésistes de type A et B

Les domaines de stabilité obtenus par simulations transitoires des SS-ADPLLs de type B sont tous les mêmes et correspondent au domaine obtenu pour un réseau de deux SS-ADPLLs de type B (défini par son équation maîtresse) représenté en Fig. 4.3. Il

est remarquable que les domaines de stabilité issus des simulations transitoires de (4.25b), ou en résolvant les LMIs du théorème 3.1 de la section 3.2.2 conduisent exactement aux mêmes résultats, indépendamment de la taille du réseau. Cela suggère que la stabilité de (4.47b) est une condition nécessaire et suffisante pour la synchronisation des réseaux cartésiens autonomes de SS-ADPLLs de type B. Par ailleurs, il est simple de vérifier que le réseau moyen (4.48b) a également le même domaine de stabilité. D'autre part, les résultats de simulation obtenus avec les SS-ADPLLs de type A dépendent de la taille du réseau : pour les petits réseaux de 2 ou 4 SS-ADPLLs, le domaine de stabilité donné par simulations transitoires coïncide effectivement avec celui calculé à partir de (4.47a). Cependant, lorsque la taille du réseau augmente, le domaine de stabilité devient plus petit que prévu par l'équation maîtresse (Fig. 4.8 - Fig. 4.10).

4.2.3.4. Comportements en régimes transitoires de réseaux de SS-ADPLLs passésistes de type A et B

Les simulations transitoires des réseaux de 4 SS-ADPLLs sont les mêmes que ceux de réseaux de 2 SS-ADPLLs et ont été donnés en Fig. 4.2 et 4.3. Tout d'abord, un réseau composé de SS-ADPLLs de type A est simulé. Les coefficients du filtre PI sont choisis de sorte que $K_1 = 0.8$ et $K_2 = -0.7$, afin d'appliquer une forte correction, tout en restant stable. L'erreur totale de chaque nœud, calculé à partir de (4.25a) est représentée à la Fig. 4.11, ainsi que celle du « réseau moyen » (4.48a), lancé à partir des mêmes conditions initiales. Les résultats montrent que, bien que les deux réseaux semblent se synchroniser dans le même laps de temps, leur comportement transitoire est assez dissemblable. Au fur et à mesure que K_1 et K_2 diminuent, les réponses des deux réseaux deviennent plus semblables (mais leur temps de stabilisation augmente).

D'autre part, la réponse d'un réseau de SS-ADPLLs de type B (4.25b) est très semblable à celle du « système moyen » correspondant (4.48b), même pour de grandes valeurs de K_1 et de K_2 . Une réponse typique est tracée sur la figure. 4.12, pour $K_1 = 0.8$ et $K_2 = -0.7$.

Ces observations et celles de la section précédente nous conduisent à la conclusion que le réseau moyen est une bonne base pour la conception des réseaux de SS-ADPLLs de type B.

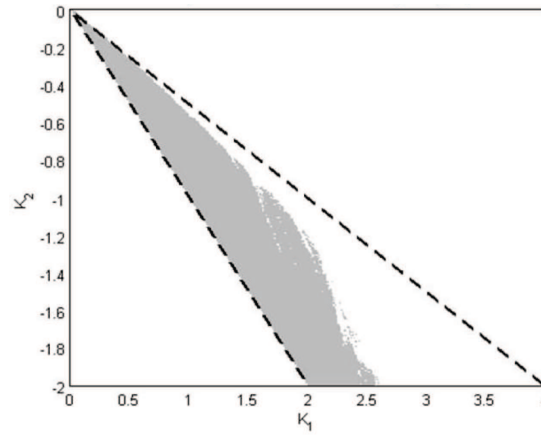


Fig. 4.8 - Domaine de stabilité d'un réseau cartésien complet de 9 SS-ADPLLs passéistes de type A. Les lignes en pointillés représentent les limites du domaine de stabilité dérivé de l'équation maitresse (4.47a).

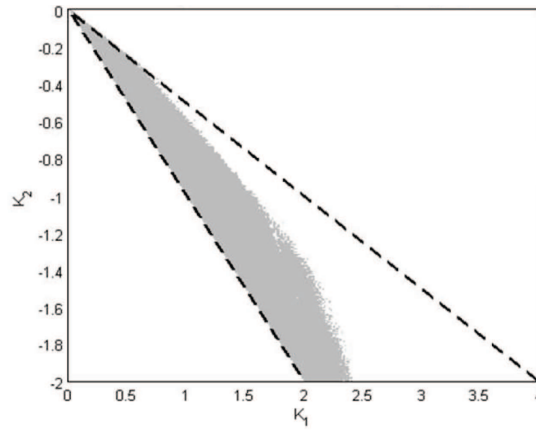


Fig. 4.9 - Domaine de stabilité d'un réseau cartésien complet de 16 SS-ADPLLs passéistes de type A. Les lignes en pointillés représentent les limites du domaine de stabilité dérivé de l'équation maitresse (4.47a).

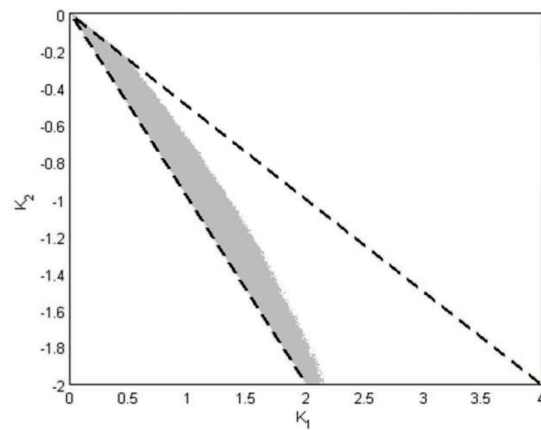


Fig. 4.10 - Domaine de stabilité d'un réseau cartésien complet de 256 SS-ADPLLs passéistes de type A. Les lignes en pointillés représentent les limites du domaine de stabilité dérivé de l'équation maitresse (4.47a).

Son comportement est très proche de celui d'un réseau auto-échantillonné (le même domaine de stabilité, un comportement similaire en régime transitoire) et il a le grand avantage d'être linéaire. Ainsi, tous les outils classiques de l'analyse linéaire (marges de stabilité, par exemple) peuvent être utilisés comme une première étape dans la conception, et leurs résultats peuvent être testés par la suite en faisant des simulations transitoires sur le réseau de SS-ADPLLs. Cette approche sera abordée plus en détail dans la section 4.3 suivante.

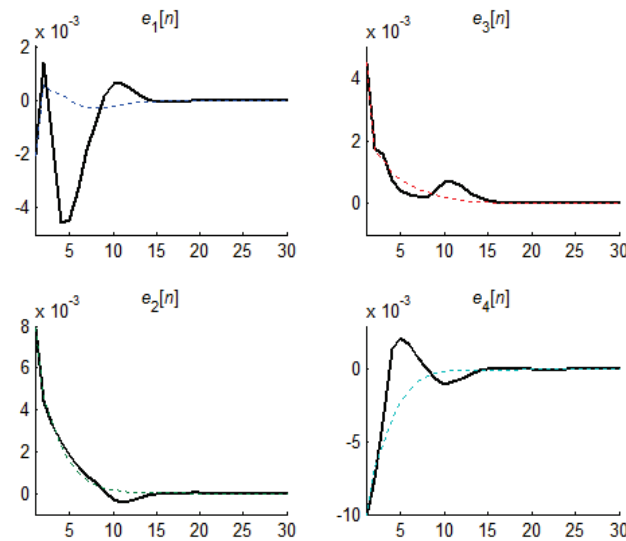


Fig. 4.11 - La réponse d'un réseau de 4 SS-ADPLLs passésistes de type A (lignes noires en gras) comparé à celle de son réseau moyen (lignes colorées en pointillés). Simulation pour $K_1=0.8$ et $K_2=-0.7$

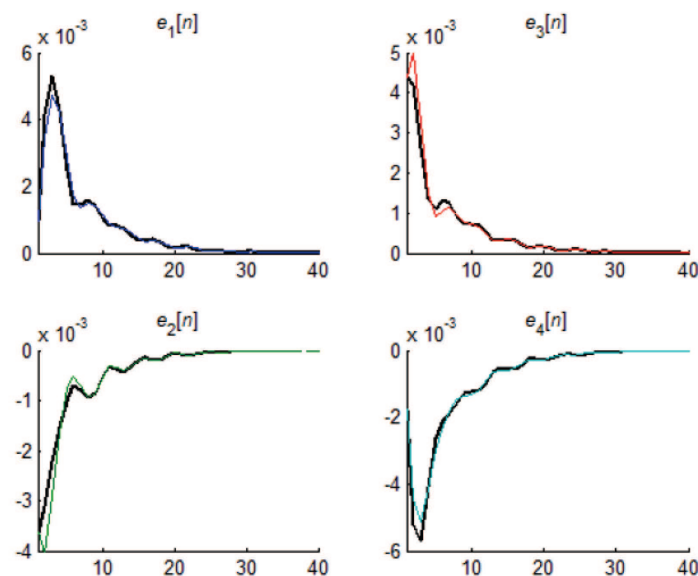


Fig. 4.12. La réponse d'un réseau de 4 SS-ADPLLs passésistes de type B (lignes noires en gras) est très similaire de celle de son réseau moyen (en couleurs). Simulation pour $K_1=0.8$ et $K_2=-0.7$

L'évolution des domaines de stabilité représentés aux Fig. 4.8-4.10 laisse à penser que plus le réseau a une taille importante, plus le domaine de stabilité est réduit pour les réseaux de SS-ADPLLs passésistes de type A. Ceci n'est cependant pas le cas pour les SS-ADPLLs passésistes de type B. Dans ce cas particulier d'implémentation du filtre numérique, la zone de stabilité trouvée est représentée à la Fig. 4.3, quel que soit le nombre de nœuds du réseau.

4.2.4. Réseaux nihilistes de tailles quelconques

Le cadre de l'étude de ces réseaux est exactement le même que celui développé à la section 4.2.2, à la seule différence que l'équation (4.18) est remplacée par :

$$\varepsilon_{lk}[n] = \begin{cases} e_{lk}[n] & \text{si } e_{lk}[n] \leq 0 \\ 0 & \text{ailleurs} \end{cases}, \quad (4.52)$$

et l'équation (4.20) par :

$$\varepsilon_{lk}[n] - \varepsilon_{kl}[n] = e_{lk}[n] \quad (4.53)$$

4.2.4.1. Équations maîtresses des SS-ADPLLs nihilistes de type A et B

L'équation maîtresse de la SS-ADPLL nihiliste de type A peut s'obtenir à partir de (4.46):

$$E[n+1] - 2E[n] + E[n-1] = -K_1 E[n] - 2K_2 E[n-1]. \quad (4.54a)$$

De même, pour la SS-ADPLL nihiliste de type B, on obtient:

$$E[n+1] - 2E[n] + E[n-1] = -K_1 E[n] - K_2 E[n-1]. \quad (4.54b)$$

On peut constater, de la même manière que pour les SS-ADPLLs passésistes que le domaine de stabilité de l'équation maîtresse des réseaux de SS-ADPLLs nihilistes de type A (resp. de type B) est le même que celui donné en Fig. 4.4 (resp. 4.5).

4.2.4.2. Domaines de stabilité des réseaux de SS-ADPLLs nihilistes de type A et B

On a représenté en Fig. 4.13, le domaine de stabilité d'un réseau cartésien de 9 SS-ADPLLs nihilistes de type A obtenu par simulation transitoire. Le domaine de stabilité est extrêmement réduit à partir de 16 SS-ADPLLs nihilistes de type A (Fig. 4.14, petit nuage de points autour des coordonnées $K_1 = 2$ et $K_2 = -0.8$) et est vide pour des réseaux de plus grande taille. Les domaines de stabilité obtenus pour des SS-ADPLLs nihilistes de type B semblent mieux se comporter et leurs domaines de stabilité semblent mieux délimités. Néanmoins, lorsque la taille du réseau augmente, la taille du domaine de stabilité diminue fortement (Fig. 4.27 à 4.29) : ceci semble indiquer que, au contraire des réseaux de SS-ADPLLs « passéistes », ceux constitués de SS-ADPLLs « nihilistes » ne sont pas extensibles à l'infini.

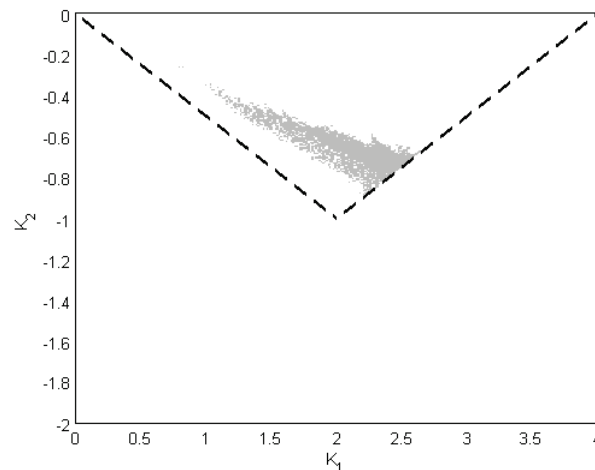


Fig. 4.13 - Domaine de stabilité (zone grise) d'un réseau de 9 SS-ADPLLs nihilistes de type A. Les lignes en pointillés représentent les limites du domaine de stabilité dérivé de l'équation maitresse (4.54a).

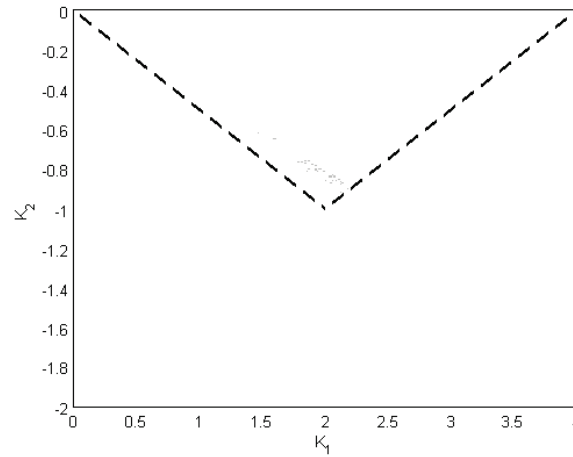


Fig. 4.14 - Domaine de stabilité (zone grise) d'un réseau de 16 SS-ADPLLs nihilistes de type A. Les lignes en pointillés représentent les limites du domaine de stabilité dérivé de l'équation maitresse (4.54a).

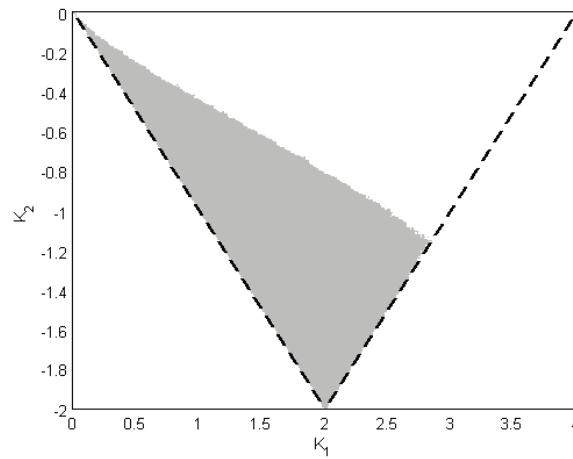


Fig. 4.15 - Domaine de stabilité (zone grise) d'un réseau de 9 SS-ADPLLs nihilistes de type B. Les lignes en pointillés représentent les limites du domaine de stabilité dérivé de l'équation maitresse (4.54b).

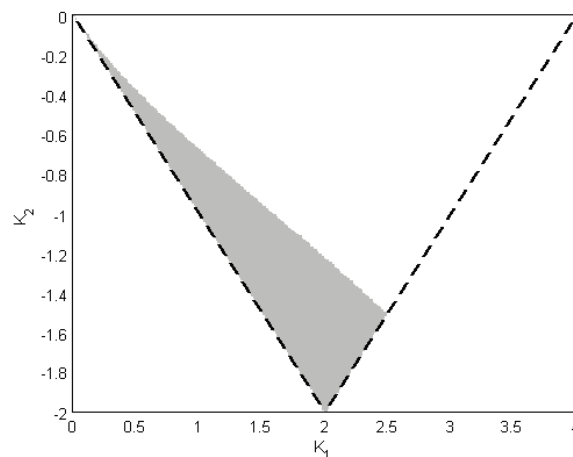


Fig. 4.16 - Domaine de stabilité (zone grise) d'un réseau de 16 SS-ADPLLs nihilistes de type B. Les lignes en pointillés représentent les limites du domaine de stabilité dérivé de l'équation maitresse (4.54b).

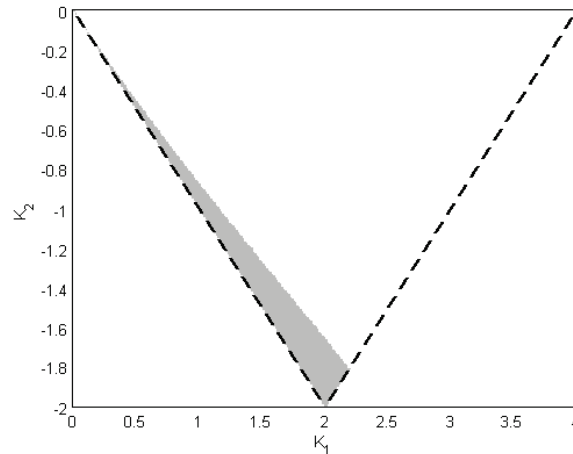


Fig. 4.17 - Domaine de stabilité (zone grise) d'un réseau de 64 SS-ADPLLs nihilistes de type B. Les lignes en pointillés représentent les limites du domaine de stabilité dérivé de l'équation maitresse (4.54b).

4.3. Les systèmes moyens comme outils d'optimisation

Dans cette partie, nous allons montrer que l'étude de la stabilité et l'analyse de la performance d'un réseau cartésien de SS-ADPLLs de type B, peuvent se faire à partir de celles du système moyen correspondant. Il suffit pour cela de construire le système moyen correspondant puis de calculer les coefficients optimums du filtre permettant de garantir la stabilité et la performance en utilisant un critère linéaire de performance tel que la marge de module. Cette démarche se justifie par le fait qu'elle permet un choix de coefficients à partir d'une approche simplifiée mais solide d'un réseau auto-échantillonné fortement non-linéaire. En effet, les liens entre un réseau de SS-ADPLLs non linéaire, son équation maîtresse et son système moyen linéaire sont donnés en Fig 4.18.

Dans notre système qui est discret, la marge de module va se caractériser par la distance entre les modules des racines du polynôme caractéristique et le cercle unité. Une conception optimale du réseau de SS-ADPLLs pourra être obtenue en choisissant les coefficients K_1 et K_2 du filtre de sorte à maximiser la marge de module du système moyen.

Rappelons les équations gouvernant l'ensemble d'un réseau de SS-ADPLLs passéiste de type B :

$$\mathbf{e}[n+1] - 2\mathbf{e}[n] + \mathbf{e}[n-1] = -\mathbf{L}(K_1\mathbf{e}[n] + K_2\mathbf{e}[n-1])$$

Le système moyen résultant est défini par :

$$\mathbf{e}[n+1] - 2\mathbf{e}[n] + \mathbf{e}[n-1] = -\mathbf{L}\left(\frac{K_1}{2}\mathbf{e}[n] + \left(\frac{K_1 + K_2}{2}\right)\mathbf{e}[n-1] + \frac{K_2}{2}\mathbf{e}[n-2]\right)$$

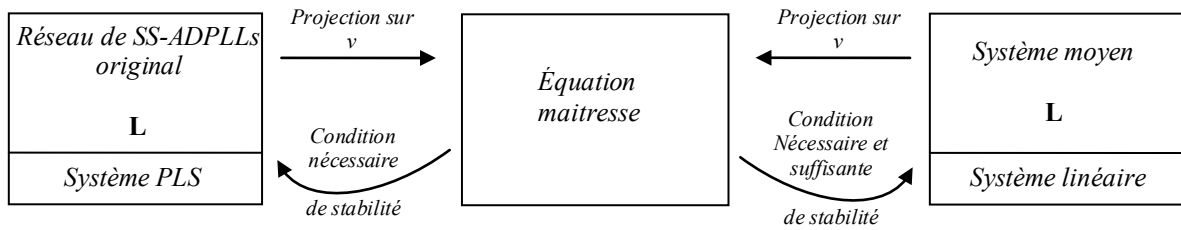


Fig. 4.18 – Position centrale de l'équation maitresse permettant d'approximer un réseau de SS-ADPLLs par son système moyen linéaire.

Les Fig. 4.19-4.20 montrent comment évoluent les coefficients optimums en fonction de la marge de module. Cette évolution suggère que les valeurs absolues des coefficients K_1 et K_2 optimales augmentent avec la taille du réseau. Cela signifie que les coefficients optimums dont il s'agit permettent aux SS-ADPLLs de verrouiller beaucoup plus vite en ce sens que ces coefficients entraînent de fortes corrections (Fig. 4.21). On remarque aussi que la marge de stabilité maximale diminue avec la taille du réseau. Cela met une limite pratique à la taille de réseau qu'on peut concevoir en garantissant la stabilité.

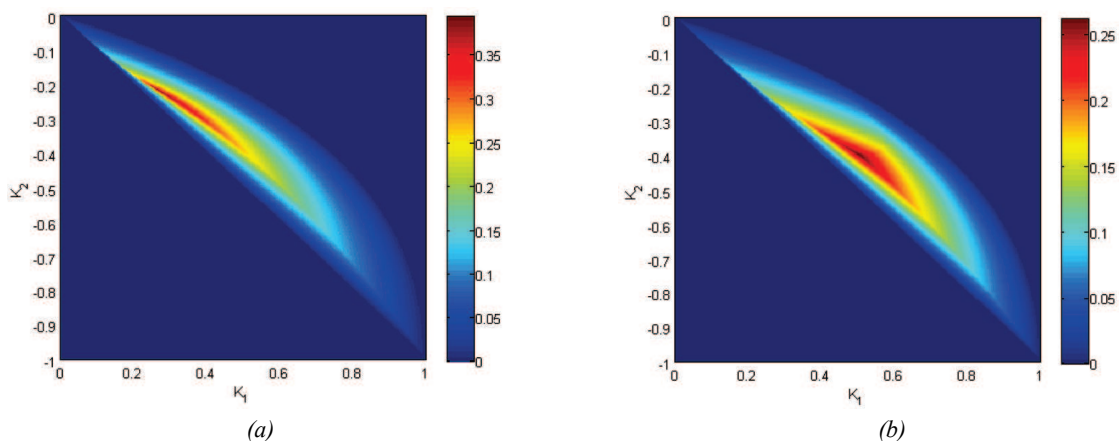


Fig. 4.19 - Domaines de stabilité du système moyen correspondant à un réseau de 2 SS-ADPLLs passésistes de type B (a) et un réseau de 4 SS-ADPLLs passésistes de type B (b) en fonction de la marge de module.

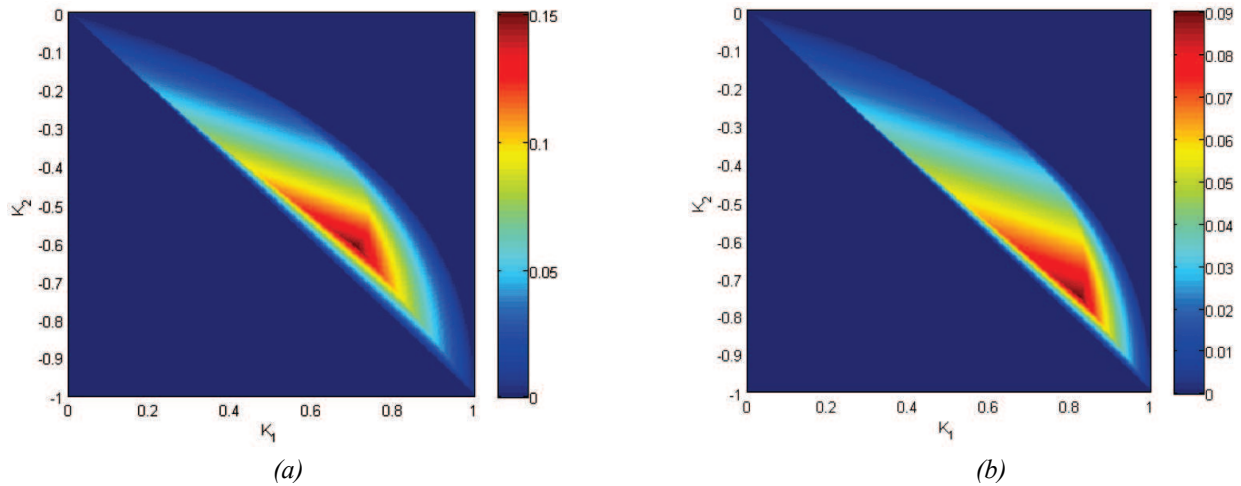


Fig. 4.20 - Domaines de stabilité du système moyen correspondant à un réseau de 9 SS-ADPLLs passésistes de type B (a) et un réseau de 16 SS-ADPLLs passésistes de type B (b) en fonction de la marge de module.

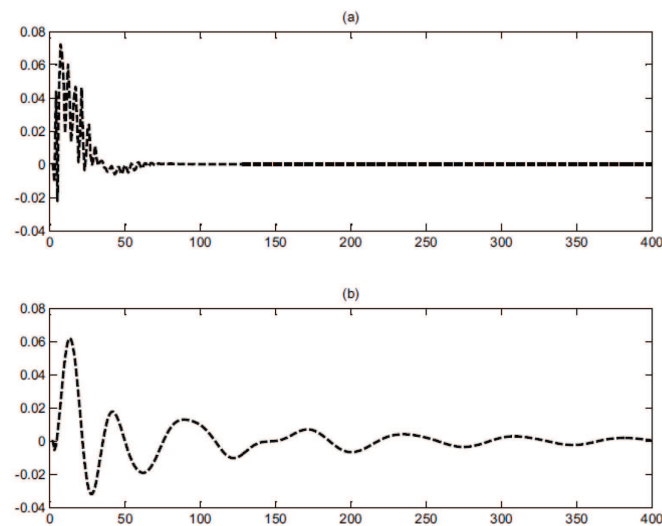


Fig. 4.21 – Erreur totale pour un nœud arbitraire d'un réseau moyen de 16 SS-ADPLLs passésistes de type B pour un couple de coefficients optimums (a) et non optimum (b).

Les résultats de Fig. 4.21 sont très semblables à ceux observés en effectuant des simulations transitoires d'un réseau auto-échantillonné de 16 SS-ADPLLs (Comme c'est notamment le cas pour 4 SS-ADPLLs, Fig. 4.12). Cependant, nous ne pouvons affirmer que la totalité des propriétés des réseaux auto-échantillonnés originaux se déduisent de celles des réseaux moyens associés. Il faudrait pour cela arriver à le démontrer d'une manière plus formelle.

Notons que l'approche qui vient d'être développée n'est valable que si l'on considère une parfaite linéarité des DPDs à l'intérieur des SS-ADPLLs. Dans le modèle réel, du fait que

la caractéristique quantifiée des comparateurs de phase de fréquence et aussi du fait de la saturation, il est nécessaire d'introduire dans la recherche des coefficients optimums, des paramètres relatifs au bruit de quantification. Nous étudions cet aspect du problème dans la section 4.4.

4.4. Sensibilité au bruit de quantification

La prise en compte du bruit de quantification qui va se faire ici a été essentiellement motivée par le fait que certains coefficients donnés optimums par l'approche des systèmes moyens traitée dans la section 4.3 se sont avérés inadaptés en les utilisant sur le système non-linéaire réel (composé de détecteurs ayant une caractéristique en escalier et saturés au bout d'un certain seuil de l'erreur temporelle e_r , voir caractéristique du DPD HODISS Fig. 2.19 de la section 2.4.1.2). En effet, dans les études précédentes, le comparateur de phase digital a été supposé infiniment linéaire. En vue de prendre au mieux en compte les non linéarités du système réel, des considérations supplémentaires vont être faites et vont permettre de considérer le fait que la caractéristique du DPD est celle d'un détecteur bang-bang [50, 66-67] pour de petites variations de l'erreur de phase autour de zéro. La démarche présentée, par souci de simplicité, est faite en ne prenant qu'en compte des SS-ADPLLs de type B dont les résultats sont plus concluants avec à la fois l'approche du système moyen tout comme celui de l'équation maitresse

Pour cela, nous allons considérer que le DPD au k^{ieme} nœud sort une erreur ε_k affectée d'un bruit comme l'illustre la Fig. 4.22 :

$$\varepsilon_k[n] = \frac{1}{|V_k|} \sum_{l \in V_k} \varepsilon_{lk}[n] + b_{lk}[n], \quad (4.55)$$

où b_{lk} représente le bruit de quantification avec :

$$b_{lk}[n] = \begin{cases} b_{lk}[n] & \text{si } b_{lk}[n] \leq 0 \\ b_{lk}[n-1] & \text{sinon} \end{cases}, \quad (4.56)$$

Pour un réseau de SS-ADPLLs de type B, l'équation du réseau moyen linéaire correspondant s'obtient de la même manière que pour l'équation (4.48b). En rajoutant à cette dernière le bruit de quantification, on obtient:

$$\mathbf{e}[n+1] - 2\mathbf{e}[n] + \mathbf{e}[n-1] = -\mathbf{L} \left(\frac{K_1}{2} \mathbf{e}[n] + \left(\frac{K_1 + K_2}{2} \right) \mathbf{e}[n-1] + \frac{K_2}{2} \mathbf{e}[n-2] \right) + K_1 \mathbf{b}[n] + K_2 \mathbf{b}[n-1] \quad (4.57)$$

où \mathbf{e} un vecteur dont la $k^{\text{ième}}$ coordonnée est e_k (tel que défini en (4.17)) et \mathbf{b} est un vecteur dont la $k^{\text{ième}}$ coordonnée est b_k :

$$b_k[n] = \frac{1}{|V_k|} \sum_{l \in V_k} b_{lk}[n].$$

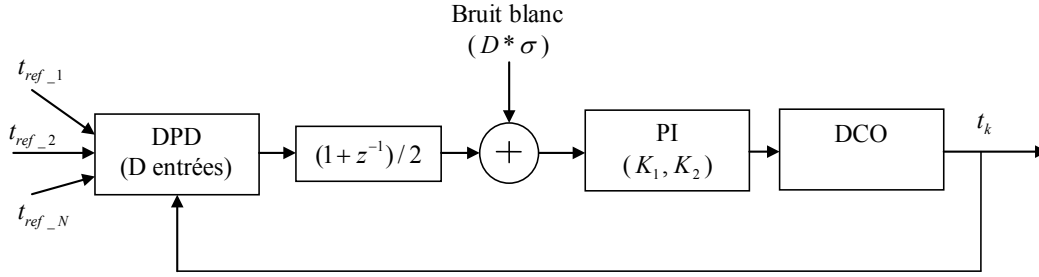


Fig. 4.22 – Modélisation de la prise en compte des erreurs de quantification dans un nœud « moyenné » de SS-ADPLL passéiste.

En faisant la transformation qui permet d'aboutir à l'équation maîtresse, on obtient :

$$E[n+1] - 2E[n] + E[n-1] = -K_1 E[n] - (K_1 + K_2) E[n-1] - K_2 E[n-2] - 2K_1 B[n] - 2K_2 B[n-1] \quad (4.58)$$

où $B[n]$ représente le bruit de chaque comparateur projeté sur le vecteur propre \mathbf{v} (4.28). Dès lors, on peut représenter l'impact qu'aura le bruit de quantification sur l'erreur totale, en utilisant la simplicité que nous offre l'équation maîtresse. Pour cela, nous allons essayer de trouver les valeurs des coefficients K_1 et K_2 qui minimisent le module de la fonction de transfert H_b entre le bruit B et l'erreur E .

$$H_b(z) = \frac{-2K_1z^{-1} - 2K_2z^{-2}}{1 - (2 - K_1)z^{-1} + (1 + K_1 + K_2)z^{-2} + K_2z^{-3}} \quad (4.59)$$

Le domaine de stabilité relatif à la valeur du module de $H_b(z)$ est donné à la Fig. 4.23 pour des réseaux de SS-ADPLLs passésistes de type B. Les couleurs sont en échelle logarithme, donc on s'intéresse ici à la ligne de niveau 0. Les coefficients optimums ainsi obtenus sont très différents de ceux suggérés à la Fig. 4.20b. Cette démarche montre assez clairement que le choix de la meilleure marge de module semble sous-optimal pour ce qui est de l'amplification du bruit de quantification. Il a été remarqué que les coefficients obtenus par la minimisation du module de H_b ont conduit à de bien meilleurs résultats sur la simulation du réseau réel (Cela sera illustré au chapitre 5).

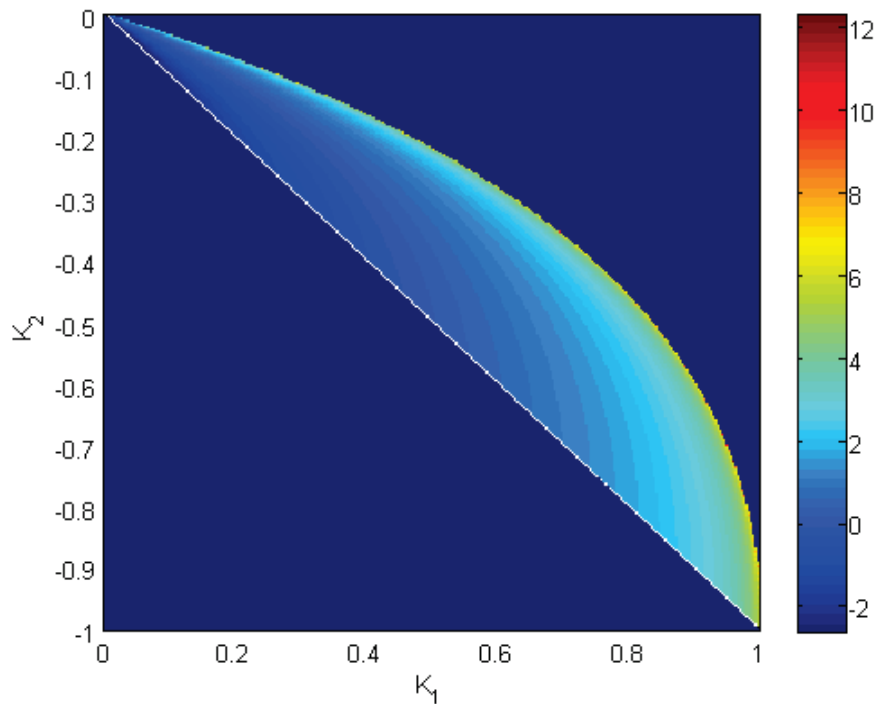


Fig. 4.23 – Module de la fonction de transfert $H_b(z)$ pour un réseau de 16 SS-ADPLLs passésistes de type B

La Fig. 4.23 permet de dire qu'il y a un compromis à trouver entre une bonne marge de module (Fig.4.20(b)) et une erreur totale faible. Si on veut construire un réseau de SS-ADPLLs avec une bonne marge de module, on est obligés de se placer à un certain point du domaine de stabilité à partir duquel, on va ensuite vérifier que l'erreur totale reste en dessous d'un certain seuil. Il faudra regarder la valeur du module de $H_b(z)$ en ce

point là et choisir une erreur de quantification acceptable en fonction de cette constatation.

4.5. Réseaux cartésiens avec référence

En pratique, le système que nous comptons mettre en œuvre fonctionne avec une référence basse fréquence (autour de 100 MHz) en entrée du réseau (Fig. 4.24), ce qui signifie qu'on fait fonctionner les oscillateurs en multiplicateurs de fréquence dans les SS-ADPLLs vu que la fréquence finale envisagée se situe autour du GHz. Le système ainsi obtenu est dissymétrique (du fait de son architecture, vu que l'un des nœuds est connectée différemment des autres) et son étude peut (*du moins en théorie*) être considérée comme une extension de l'étude des réseaux autonomes. Les résultats de simulations ont montré que les domaines de stabilité obtenus avec des réseaux comportant une référence sont les mêmes que ceux des réseaux autonomes. La seule différence réside dans le fait qu'avec une référence, la fréquence finale de tous les oscillateurs (divisée par le rapport de multiplication) se cale sur celle de la référence (la consigne). Ceci s'explique par le fait que la référence étant de fréquence fixe, tous les oscillateurs s'auto-corrigent par rapport à cette fréquence jusqu'à se caler sur elle. Pour les réseaux autonomes par exemple, la fréquence finale est difficilement prévisible car elle dépend de la valeur exacte des conditions initiales et des coefficients du filtre de boucle.

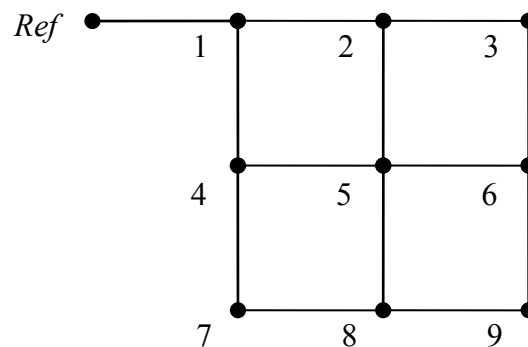


Fig. 4.24 – Réseau cartésien de 9 SS-ADPLLs avec une référence

4.6. Conclusion

L'étude de la stabilité d'un réseau de distribution d'horloge basée sur des SS-ADPLLs a été réalisée dans ce chapitre. Plusieurs approches ont été utilisées pour déterminer la stabilité de ces réseaux. Celle basée sur des simulations transitoires du réseau, conduit à des résultats qui peuvent être dépendantes des conditions initiales du système. Les approches consistant à déterminer l'équation maitresse du réseau ou l'utilisation du système moyen peuvent paraître assez limitées: elles reposent sur une topologie de réseau particulier (cartésien), et sont valables pour les réseaux autonomes seulement. Les conditions de stabilité obtenues sont nécessaires (pas suffisantes). Toutefois, elles fournissent un outil très simple permettant de déterminer, comme point de départ, les limites du domaine dans lequel les coefficients du filtre doivent être choisis pour assurer la stabilité. Par ailleurs, notre étude indique que les propriétés (pas seulement la stabilité, mais aussi le temps d'établissement du régime permanent) des réseaux cartésien de SS-ADPLLs de type B peuvent sans risque être dérivées de celles du réseau moyen correspondant. Une démonstration rigoureuse de cette propriété reste encore à établir. Enfin, une analyse linéaire fut établie pour déterminer les coefficients optimums en tenant compte du bruit de quantification en vue de nous rapprocher des contraintes réelles de notre système.

Chapitre 5 :

Validation des résultats théoriques

5.1. Introduction

Bien que la SS-ADPLL utilisée dans le projet ait été précisément décrite dans la section 2.3 du chapitre 2, son étude a été basée sur un modèle haut-niveau dont les équations ont été codées sous MATLAB afin d'en étudier le comportement. Toutes les simulations ont été effectuées jusqu'à présent en ne prenant en compte que ces équations idéalisées, et en négligeant la quantification et la saturation du DPD. En vue d'apporter une validation des précédents résultats théoriques, nous allons cette fois ci, d'abord prendre en compte les deux phénomènes non-linéaires précités, dans des simulations transitoires. Enfin ces tests vont être comparés à ceux réalisés chez nos partenaires du LIP6, sur une plateforme FPGA sur laquelle a été implémenté un réseau cartésien de 16 nœuds de SS-ADPLLs. Contrairement aux simulations précédentes où il avait été possible d'effectuer des cartographies très précises des domaines de stabilité moyennant des maillages très fins sur le choix des différents coefficients de filtres, nous nous contenterons sur le modèle FPGA, de sélectionner un jeu de coefficients de sorte à pouvoir tester de manière succincte, la stabilité vu que chaque couple de coefficient de filtre à tester demande une reprogrammation de la plateforme FPGA.

5.2. Simulation de réseaux passésistes en présence de non-idéalités

Les SS-ADPLLs passésistes, en particulier celle de type B, sont à quelques considérations près (comme par exemple les phénomènes de chevauchement ou perte de cycle¹), celles qui sont les plus similaires de ceux développées dans le projet. C'est la raison pour laquelle nous allons nous intéresser exclusivement à cette famille de SS-ADPLLs dans cette partie. Les simulations pour les SS-ADPLLs de type A sont données en

¹ Le chevauchement est le phénomène par lequel, en considérant par exemple un nœud de SS-ADPLL avec une horloge de référence, deux fronts successifs de l'horloge de référence (resp. locale) sont détectés alors que le front de l'horloge locale (resp. de référence) n'est pas encore détectée due à son retard.

annexe A.3. L'erreur temporelle maximale des DPDs est fixée à 12% de la période du signal jouant le rôle de référence conformément aux possibilités standards de variation de la fréquence d'un DCO autour de sa fréquence centrale dans un cadre réel [33] et des paramètres fournies par le LIP6. Les périodes centrales des DCOs dans nos réseaux étant centrées autour de la période unité pour toutes les simulations, on définit le pas de quantification Q des DPDs suivant la caractéristique donnée en Fig. 2.19 comme suit :

$$Q = \frac{12\%}{15} = 8 \times 10^{-3}, \quad (5.1)$$

La saturation est donnée par :

$$S = 16 \times Q = 0.128 \quad (5.2)$$

En prenant en compte la saturation et la quantification, la sortie du DPD du $k^{\text{ième}}$ nœud au $n^{\text{ième}}$ front montant de l'horloge locale $t_k[n]$ est égale à:

$$\varepsilon_k[n] = \frac{1}{|V_k|} \sum_{l \in V_k} \varepsilon_{lk}[n],$$

$\varepsilon_{lk}[n]$ étant l'erreur quantifiée.

Le code utilisé pour effectuer les simulations transitoires prenant en compte les non-linéarités des réseaux est donné en annexe A.5. Pour les réseaux de petites tailles (réseaux de 4, 9 et 16 SS-ADPLLs), 300 valeurs de K_1 sur l'axe des abscisses et 300 autres pour K_2 sur l'axe des ordonnées permettent de définir une grille de 90000 coordonnées (K_1, K_2) pour lesquelles les simulations sont lancées dans une boucle permettant d'effectuer 3000 itérations (nombre de fronts montants maximum). Pour les réseaux de 81 et 256 SS-ADPLLs, dû au temps de calcul important, le maillage est plus grossier (100 valeurs de K_1 sur l'axe des abscisses et 100 autres pour K_2 sur l'axe des ordonnées) mais permet cependant d'avoir un bon aperçu du domaine de stabilité. Les cartographies précises représentant les domaines de stabilité pour différents réseaux de SS-ADPLLs sont ainsi obtenus. Ces domaines sont affichés de sorte à pouvoir apprécier le gradient de l'erreur temporelle en calculant la moyenne des valeurs absolues de toutes les erreurs totales sur les 100 derniers échantillons. En effet, cela permet de localiser les couples de coefficients donnant l'erreur temporelle finale la plus faible

(\pm une fois le pas de quantification ou effet bang-bang en zéro) et ceux donnant l'erreur temporelle la plus importante (on affiche sur les cartes que des erreurs inférieures à 10 fois le pas de quantification).

5.2.1. Réseaux passésistes de type B

Les domaines de stabilité pour des réseaux de SS-ADPLLs de type B sont donnés de la Fig. 5.1 à la Fig. 5.5. Les résultats obtenus valident la condition nécessaire de stabilité (définie par l'équation maitresse) obtenue en section 4.2.2 pour les SS-ADPLLs de type B. On note aussi le fait que la taille des domaines de stabilité de ces types de réseaux diminue sensiblement lorsque la taille du réseau augmente. Ces résultats, plus réalistes, diffèrent très sensiblement de ceux obtenus pour les réseaux passésistes de type B idéalisés (suggérant une conservation de la taille du domaine de stabilité avec la taille du réseau) et met en évidence l'effet de la quantification sur les domaines de stabilité.

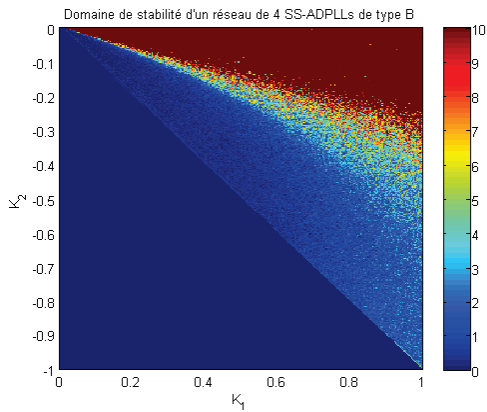


Fig. 5.1 – Domaine de stabilité d'un réseau de 4 SS-ADPLLs passésistes de type B.

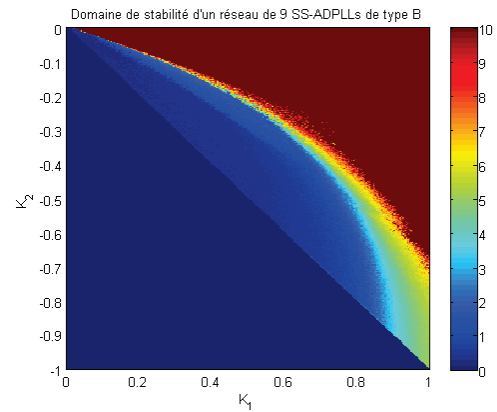


Fig. 5.2 – Domaine de stabilité d'un réseau de 9 SS-ADPLLs passésistes de type B.

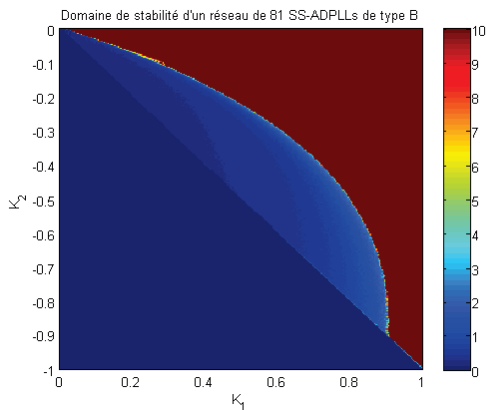


Fig. 5.3 – Domaine de stabilité d'un réseau de 81 SS-ADPLLs passésistes de type B.

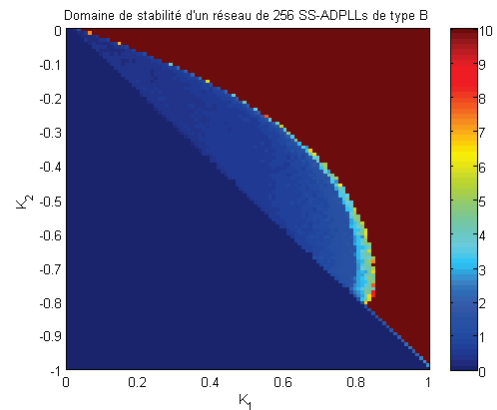


Fig. 5.4 – Domaine de stabilité d'un réseau de 256 SS-ADPLLs passésistes de type B.

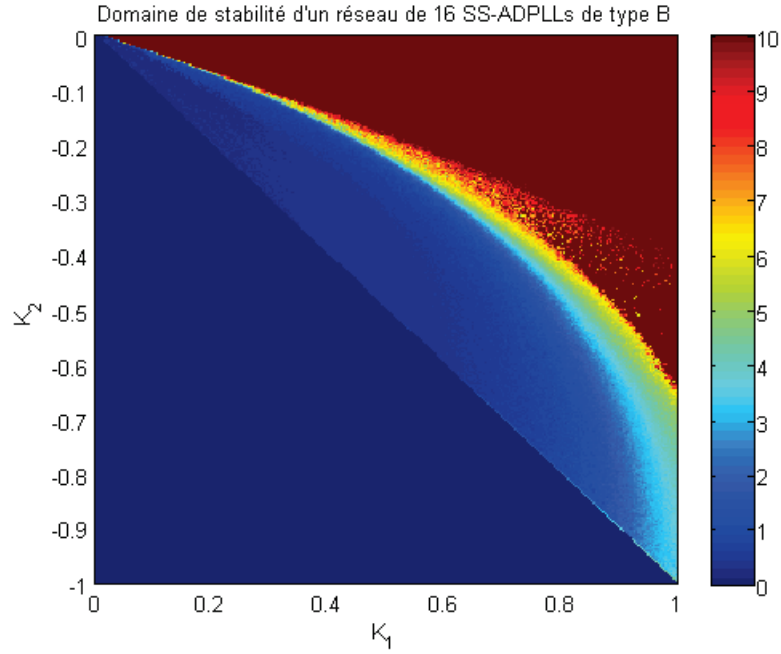


Fig. 5.5 – Domaine de stabilité d'un réseau de 16 SS-ADPLLs passésistes de type B.

En effet, on peut observer la cohérence des résultats actuels avec ceux obtenus à la Fig. 4.23 de la section 4.4 qui traite justement de la sensibilité des réseaux au bruit de quantification.

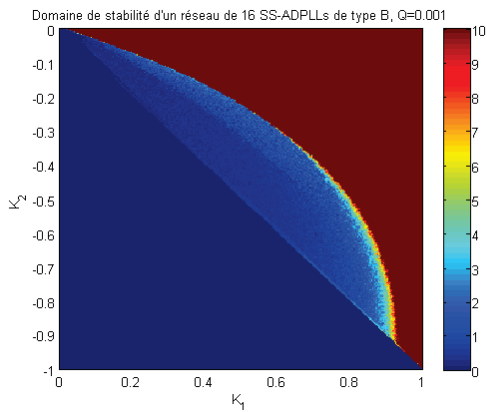


Fig. 5.6 – Domaine de stabilité d'un réseau de 16 SS-ADPLLs passésistes de type B avec $Q=0.001$.

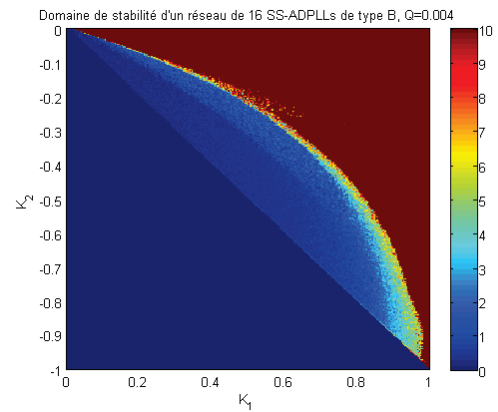


Fig. 5.7 – Domaine de stabilité d'un réseau de 16 SS-ADPLLs passésistes de type B avec $Q=0.004$.

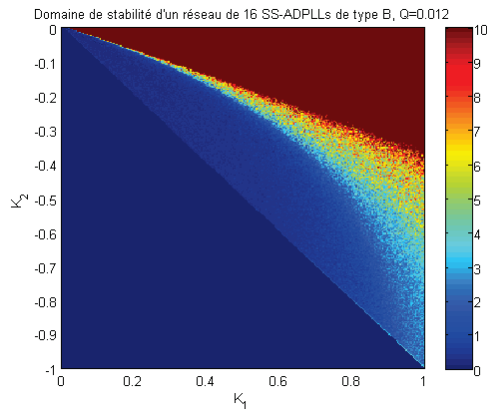


Fig. 5.8 – Domaine de stabilité d'un réseau de 16 SS-ADPLLs passésistes de type B avec $Q=0.012$.

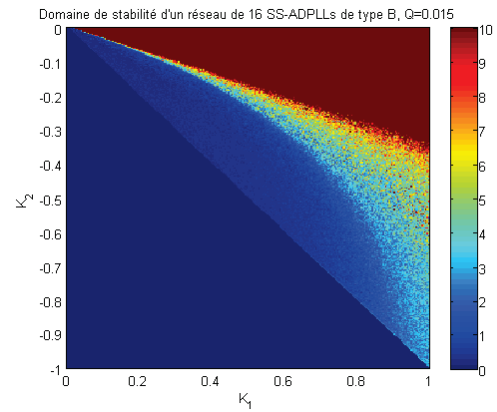


Fig. 5.9 – Domaine de stabilité d'un réseau de 16 SS-ADPLLs passésistes de type B avec $Q=0.015$.

Les figures suivantes (Fig. 5.6-5.9), montrent comment évolue le domaine de stabilité d'un réseau de 4 SS-ADPLLs de type B en fonction du choix du pas de quantification, la valeur de la saturation étant fixée à 16 fois le pas de quantification. Elles permettent de voir que le domaine de stabilité générant les erreurs totales les plus faibles se conserve globalement. Lorsque le pas de quantification augmente, cela a pour effet d'élargir la zone intermédiaire entre le domaine pouvant être considéré comme stable (valeur absolue de la moyenne des erreurs totales inférieures à un pas de quantification) et le domaine considéré comme étant celui du maximum de l'instabilité (valeur absolue de la moyenne des erreurs totales supérieure ou égale à 10 fois le pas de quantification). Ce phénomène peut vraisemblablement s'interpréter comme l'entrée du réseau dans des cycles limites stables. En effet, si l'on prend l'exemple de la Fig. 5.9 où le pas de quantification Q est fixé à 0.015 et où la saturation vaut donc $S=16*Q=0.24$, on se rend compte que l'erreur maximale temporelle des DPDs vaut 22.5% de la période du signal de référence. Dans cet exemple précis, cela vaudrait dire que l'on augmente les possibilités de variation de fréquence des DCOs autour de leur fréquence nominale. Ce dernier paramètre, augmentant les degrés de liberté du réseau et donc sa complexité permet l'entrée du système non-linéaire dans un état stable mais où toutes les erreurs ne convergent pas individuellement vers des valeurs faibles car les périodes des DCOs ne sont pas suffisamment proches pour permettre la convergence du réseau vers l'état d'équilibre stable désiré [1]. C'est d'ailleurs la raison pour laquelle Pratt et N'Guyen ont préconisé (pour éviter ces états d'équilibre stables indésirables) dans [1] d'utiliser des comparateurs ayant une pente négative après une erreur de phase supérieure à 90° .

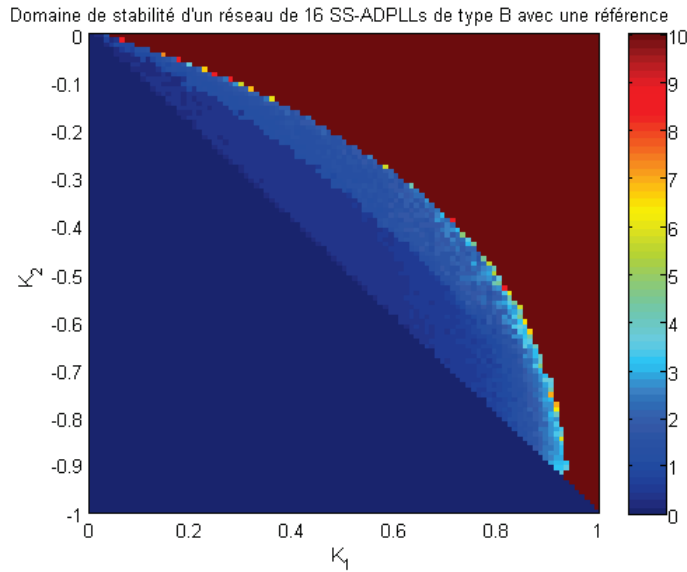


Fig. 5.10 – Domaine de stabilité d'un réseau de 16 SS-ADPLLs passésistes de type B avec une référence.

La Fig. 5.10 représente le domaine de stabilité obtenu en utilisant l'un des nœuds du réseau de 16 SS-ADPLLs de type B comme référence. Le domaine de stabilité obtenu est assez similaire à celui obtenu pour le réseau autonome à la Fig. 5.5. La différence essentielle entre les deux cas est le fait que dans un réseau avec une référence, la convergence du réseau se fait sur la référence, c'est-à-dire que toutes les périodes finales des DCOs du réseau approcheront celle de la référence de manière asymptotique (Fig. 5.11 et Fig. 5.12 pour $K_1 = 0.1866$ et $K_2 = -0.1362$). Pour un réseau autonome par contre, la convergence des périodes des DCOs se fait sur une valeur aléatoire dépendant essentiellement des conditions initiales du système. Les différentes simulations transitoires effectuées ont permis de constater que cette la valeur finale de la période des DCOs pouvait tendre vers une valeur fixe s'approchant de la moyenne des périodes nominales des DCOs du réseau. Cependant, il a été observé que dans certains cas, l'ensemble des périodes pouvant converger vers une valeur indéfinie, dont la valeur absolue pouvait croître de façon monotone, laissant penser à une divergence du réseau.

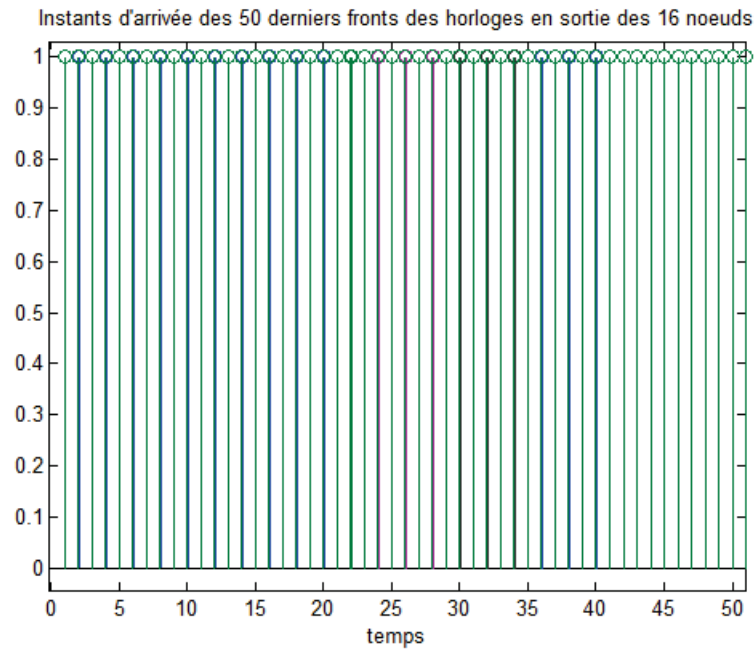


Fig. 5.11 – Derniers fronts (50) des horloges en sorties d'un réseau de 16 SS-ADPLLs de type B.

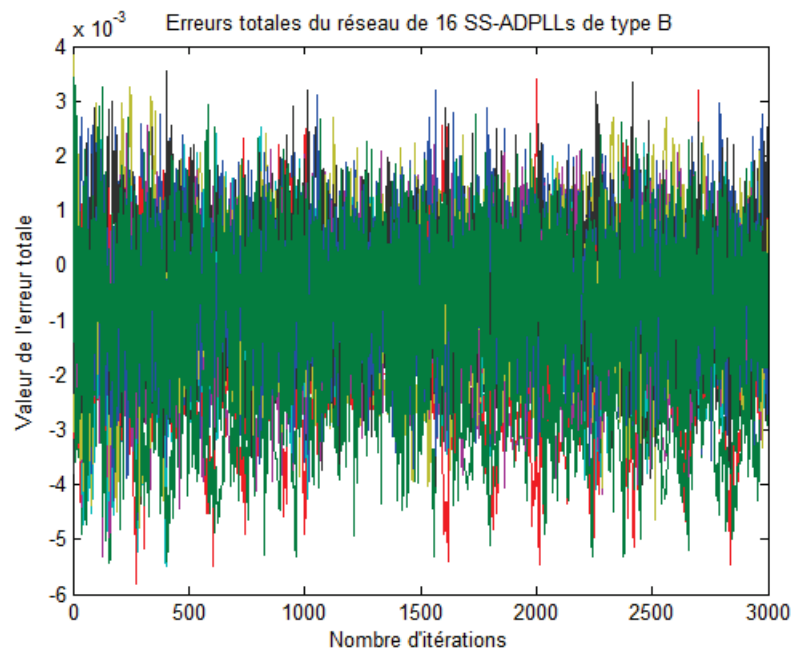


Fig. 5.12 – Erreurs totales en sortie d'un réseau de 16 SS-ADPLLs de type B avec référence.

5.3. Prototype FPGA d'un réseau de SS-ADPLLs

Un prototype FPGA d'un réseau de PLLs a été réalisé au LIP6 pour nous permettre, en reproduisant fidèlement les blocs fonctionnels du réseau (ASIC) de SS-ADPLLs actuellement en train d'être conçu au sein du LIP6, d'observer le comportement réel du système et de valider les théories développées pour l'étude de la stabilité afin d'en optimiser le fonctionnement. L'émulateur FPGA a de ce fait la même architecture et les mêmes paramètres que l'ASIC (remis à l'échelle proportionnellement à la fréquence du FPGA qui est limitée).

Dans toutes les simulations transitoires précédentes, les équations des SS-ADPLLs avaient toutes été normalisées. Ainsi, pour faire en sorte de reproduire fidèlement le comportement des réseaux de SS-ADPLLs avec leurs paramètres ASIC [62] et tenir compte de la quantification du DPD, le pas de quantification de celui-ci va être recalculé. Le tableau 5.1 donne dans sa colonne gauche les paramètres ASIC des SS-ADPLLs, celle du milieu ceux du FPGA et dans celle de droite, ceux des paramètres normalisés utilisés dans les simulations. Comme on peut le noter en observant le tableau, le réseau ASIC comporte un diviseur de fréquence par 8 car la fréquence nominale du DCO réalisé en ASIC [62] est de 2GHz.

	Paramètres ASIC	Paramètres FPGA	Paramètres normalisés
Fréquence de référence	$F_r = 250 \text{ MHz}$	$F_r = 50.04 \text{ KHz}$	$F_r = 1 \text{ Hz}$
Fréquence nominale du DCO	$F_i = 2 \text{ GHz}$	$F_i = 200 \text{ KHz}$	$F_i = 1 \text{ Hz}$
Facteur de division	$M = 8$	$M = 4$	$M = 1$
Pas de quantification	$\tau_{TDC} = 32 \text{ ps}$	$\tau_{TDC} = 149 \text{ ns}$	$\tau_{TDC} = 8 \text{ ms}$
Gain du DCO	$g = 200 \text{ KHz/LSB}$	$g = 40.03 \text{ Hz/LSB}$	$g = 1 \text{ Hz/LSB}$

Tableau 5.1 – Tableau récapitulatif des paramètres ASIC, FPGA et des paramètres normalisés.

5.3.1. Présentation de l'architecture du système testé

Le fonctionnement de certains blocs de l'implémentation ASIC étant basé sur la physique des retards et ces caractéristiques de conception n'étant pas fournis pour la plupart des plateformes FPGA existantes, on fait fonctionner le DCO et le TDC en mode de signaux mixtes¹. A cause des limitations fréquentielles de la carte FPGA, sa fréquence de fonctionnement est K fois inférieure à celle de l'implémentation CMOS [68, 95]. Ainsi tous les paramètres du système ainsi que les unités de temps doivent être multipliés par K . Ces paramètres sont la fréquence centrale du DCO, son pas de fréquence et la résolution du TDC donnés dans le tableau 5.1. On peut déduire à partir de ce même tableau, le facteur d'adaptation des coefficients du filtre pour le prototype FPGA.

$$G = \tau_{TDC} \cdot g \cdot T_i^2 \approx 0.1 \quad (5.3)$$

où τ_{TDC} est le gain du DPD et g celui du DCO. Les coefficients optimaux du filtre de boucle d'un réseau de SS-ADPLLs de type B tels que définis par la méthode de la section 4.4 étant normalisés, il suffit de les diviser par le facteur G pour obtenir les bons coefficients relatifs aux paramètres FPGA ici appliqués. Un couple de coefficients optimaux (selon la mise en forme du bruit de quantification) normalisés de la Fig. 4.23 sont :

$$K_1 = 0.08 \text{ et } K_2 = -0.0797$$

Les coefficients optimaux qui doivent être appliqués au prototype FPGA sont donc :

$$K_1 = 0.8 \text{ et } K_2 = -0.797$$

Ces coefficients déterminés par la théorie et validés par des simulations transitoires sous MATLAB vont être utilisés pour synchroniser le réseau de 16 SS-ADPLLs implémenté sous FPGA comme le montrent les mesures effectuées sur le système FPGA.

¹ On utilise à la fois des signaux numériques et analogiques par le biais d'interfaces constitués par des convertisseurs analogiques numériques (CAN) et de convertisseurs numériques analogiques (CNA).

5.3.2. Mesures expérimentales

Le prototype FPGA a été implémenté sur une plate-forme CYCLONE II EP2C70. Les signaux sont capturés par un analyseur logique et sont ensuite affichés sur un oscilloscope. On affiche sur les Fig.5.13-Fig.5.15 les horloges des 16 SS-ADPLLs et l'horloge de référence. D'abord, le réseau est configuré pour fonctionner en mode bidirectionnel (chaque nœud transmet une information d'erreur de phase à ses plus proches voisins et reçoit aussi de ceux-ci des informations d'erreurs de phase dans configuration cartésienne du réseau), et les fréquences centrales des DCOs sont pareilles pour toutes les SS-ADPLLs. On peut remarquer dans la Fig. 5.13 que le réseau est stable et il n'y a pas de *modelock*¹.



Fig. 5.13 – Signaux de sortie d'un réseau de 16 SS-ADPLLs, on n'observe pas de modelock (configuration : bidirectionnelle, fréquences centrales identiques).

Élimination des modes indésirables

Précédemment, il a été choisi pour le réseau de 16 SS-ADPLLs, des fréquences centrales identiques pour chaque nœud du réseau. Lorsque celles-ci sont différentes, la configuration du réseau restant la même, l'on a pu cependant observer des phénomènes

¹ Le modelock correspond à un état stable du réseau où toutes les erreurs totales de tous les nœuds sont nulles mais où il peut persister des erreurs individuelles non nulles entre une ou plusieurs mailles du réseau.

de *modelocks*. En effet, bien que les 16 SS-ADPLLs puissent suivre les variations de la fréquence de référence, des erreurs de phase individuelles ont été notées dans le réseau.

La configuration unidirectionnelle (configuration chaîne du réseau, l'erreur de phase se transmettant dans une seule direction) du réseau peut être utilisée pour tenter d'éliminer les modes stables indésirables. Lors de la phase de démarrage du système, l'idée qui a été exploitée fut de lancer le réseau sur son mode unidirectionnel ; lorsque le régime transitoire est terminé, on active le mode bidirectionnel. En ce moment, les nœuds sont entièrement interconnectés avec leurs voisins. Les petites erreurs persistantes dans le mode unidirectionnel n'empêchent pas au système d'arriver dans le mode souhaitable (synchronisation) lorsque le lien bidirectionnel est établi. On appelle cette procédure une *configuration dynamique*. L'efficacité de cette façon de procéder est illustrée dans les Fig.5.14 et Fig. 5.15. Dans un premier temps, en mode unidirectionnel, le réseau atteint l'état d'équilibre stable, bien que les erreurs de phase ne soient pas suffisamment faibles, notamment celles pour les nœuds en bout de chaîne. Cependant, on peut garantir qu'il n'y a pas de modelock.

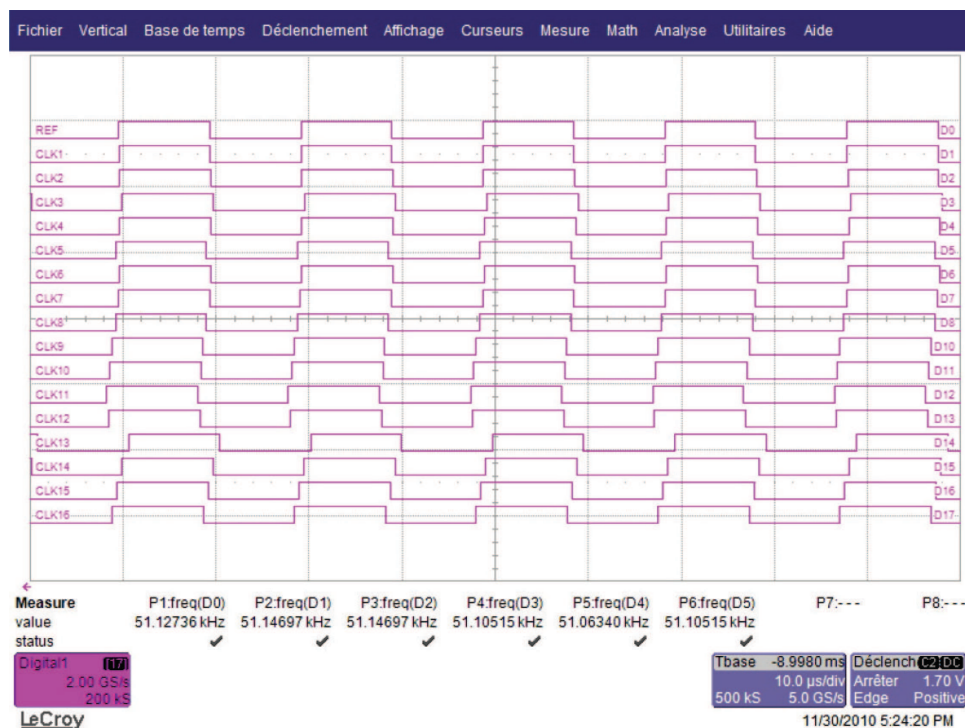


Fig. 5.14 – Signaux de sortie d'un réseau de 16 SS-ADPLLs, phase initiale de la configuration dynamique (mode unidirectionnel), fréquences centrales différentes, pas de modelock.

Ensuite, dans la deuxième phase qui consiste à faire fonctionner le système de façon bidirectionnelle. Les erreurs de phase sont beaucoup plus petites.

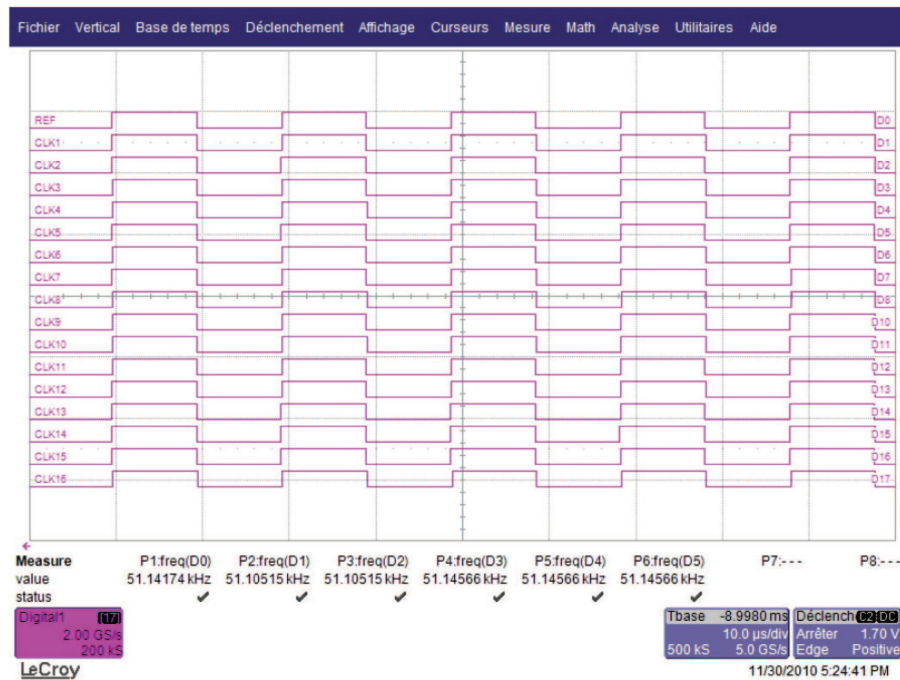


Fig. 5.15 – Signaux de sortie d'un réseau de 16 SS-ADPLLs, phase secondaire de la configuration dynamique (mode bidirectionnel), fréquences centrales différentes, pas de modelock et erreurs plus petites.

5.4. Conclusion

Cette partie s'est consacrée au travers de simulations avancées à la validation des résultats théoriques obtenus des modèles de SS-ADPLLs. Ces simulations ont pris en compte les caractéristiques non-linéaires du système physique réel en intégrant ces derniers paramètres dans les équations du système. Un modèle Simulink de SS-ADPLL a aussi été testé en vue d'intégrer dans les simulations les formes d'ondes des signaux d'entrées / sortie du système. Enfin, un réseau de générateurs d'horloges de 16 nœuds composé de SS-ADPLLs, implémenté sur une plateforme FPGA par nos partenaires du LIP6, a permis de tester les coefficients donnés stables et ainsi d'apporter une validation expérimentale de ce type de système. La conception effectuée eut aussi pour objectif de permettre aux concepteurs de choisir les coefficients de filtre permettant non seulement d'assurer la synchronisation de réseaux cartésiens de SS-ADPLLs mais aussi de pouvoir assurer des spécifications de performances. Différents tests ont été effectués sur ce

prototype en balayant le domaine de stabilité défini par la théorie développée au chapitre 4. Ces tests ont été concluants ainsi que la solution proposée et mise en œuvre en vue d'éviter les modes stables indésirés. Les résultats obtenus sur le prototype FPGA permettent d'être optimistes quant à une implémentation CMOS future.

CONCLUSIONS GÉNÉRALES ET PERSPECTIVES

Dans le vaste domaine des sciences de l'information qui comprennent l'informatique, les télécommunications et le traitement du signal, le temps est une notion omniprésente. Les signaux manipulés et échangés qui peuvent être des données informatiques circulant à travers un réseau informatique ou au sein d'un ordinateur dépendent effectivement du temps. Le réglage des horloges et plus particulièrement leur synchronisation (mécanisme permettant à deux ou plusieurs systèmes distincts d'avoir une différence entre leurs temps subjectifs la plus faible possible) est une propriété importante qui a été le sujet principal de cette thèse. La mesure du temps dans les dispositifs électroniques est traditionnellement faite localement à l'aide d'horloges électroniques pilotées par des quartz. Même si ces fréquences sont en général très précises, on y observe des fluctuations dues à de nombreux facteurs (par exemple aux variations de température ou à l'usure des composants électroniques) et des différences de phase. Les fluctuations entraînent des dérives aléatoires plus ou moins importantes (jusqu'à plusieurs secondes en fin de journée) d'autant plus que les fréquences de fonctionnement sont hautes (système de multiplication de fréquence permettant d'atteindre l'ordre du Gigahertz). En effet, le temps étant mesuré en comptant les impulsions d'horloge, les petites fluctuations sur la fréquence d'horloge peuvent conduire à des erreurs de temps importantes, dès lors que celles-ci sont accumulées sur un grand nombre d'impulsions. La démarche proposée dans les travaux ici présentés consiste à mettre en réseau plusieurs oscillateurs (horloges) et à les synchroniser par un système de verrouillage de phase permettant d'effectuer une correction dynamique à chaque nœud du réseau en l'asservissant sur la phase de ses voisins. Une étude théorique approfondie du système d'horlogerie constitué par le réseau de SS-ADPLLs couplées, rendue plus complexe par la nature discrète des blocs composant le système fut établie.

La première contribution de cette thèse réside dans la manière dont les équations régissant le fonctionnement des SS-ADPLLs ont été posées. En effet, la combinaison des équations des différents blocs du système en une seule équation semblable à celle d'une suite récurrente linéaire avec comme terme principal l'erreur de phase (ou l'erreur temporelle) a permis de simplifier de manière très significative l'étude du comportement d'une ADPLL et d'une SS-ADPLL en particulier. Les différents modes de fonctionnement d'une SS-ADPLL, d'abord basés sur le type de prédiction choisi pour l'erreur de phase et ensuite dans le design du filtre digital ont ainsi pu être introduits en une équation, permettant entre autre d'analyser la façon dont le verrouillage se fait, l'état synchrone et les conditions de stabilité de ce genre de système. Ainsi, la synchronisation pour un nœud de SS-ADPLL fut aisément étudiée autant que la synchronisation pour des réseaux cartésiens quelconques de SS-ADPLLs.

Les équations régissant le fonctionnement des SS-ADPLLs étant celles de systèmes linéaires par morceaux, la seconde contribution de ce travail fut d'appliquer des outils classiques de l'automatique pour étudier la stabilité des ces SS-ADPLLs en recherchant les fonctions de Lyapunov quadratiques par morceaux (PQLF). Ces fonctions permettent de déterminer les coefficients du filtre permettant d'assurer la

stabilité et donc la synchronisation en résolvant un système d'inégalités matricielles. Cette dernière méthode bien qu'étant rigoureuse donne des résultats très conservatifs et ne peut pas être appliquée à toutes les catégories de SS-ADPLLs. D'autre part, c'est une méthode qui ne permet d'obtenir que des conditions suffisantes de synchronisation.

Une dernière et importante contribution de cette thèse fut la mise en évidence de façon rigoureuse de l'existence d'une équation maîtresse régissant tous les réseaux quadratiques de SS-ADPLLs. Cette équation maîtresse variant en fonction du type de SS-ADPLL dont il est question permet de dériver une condition nécessaire de stabilité et de synchronisation quelque soit la taille du réseau de SS-ADPLLs. De plus, nous avons pu établir une relation entre les systèmes moyens de réseaux de SS-ADPLLs et leurs équations maîtresses. Les systèmes moyens permettent de trouver des conditions suffisantes de synchronisation, ce qui signifie que pour des coefficients de filtre suffisamment faibles, le système peut être équivalent au modèle continu.

En vue de valider les résultats théoriques obtenus, des simulations transitoires ont été effectuées sous Matlab d'abord sur des modèles de SS-ADPLLs idéalisés et ensuite en prenant en compte les toutes les non idéalités (quantification et saturation) des réseaux. Fort des résultats obtenus sur l'ensemble des études et des simulations réalisées, on peut prétendre disposer d'outils d'analyse et de simulation pour la conception des systèmes d'horlogerie distribuée à base de réseaux de SS-ADPLLs.

Cependant, les méthodes et outils proposés peuvent être améliorés. En ce qui concerne les méthodes, il serait intéressant d'étendre le modèle de SS-ADPLL actuel en prenant en compte les phénomènes de chevauchement ou d'autres non linéarités présentes sur des systèmes physiques réels tels que saturation du DCO. De plus, les perturbations sur les SS-ADPLLs peuvent ne pas exclusivement être causées par le bruit de quantification mais aussi par la quantification de la fréquence du DCO [84]. Ce dernier aspect de non-linéarité mériterait aussi une étude plus approfondie de sorte à en connaître les différents effets sur les domaines de stabilité ou les comportements des réseaux en régime transitoire. D'autre part, le domaine de stabilité obtenu en utilisant les PQLFs pourrait être réétudié à appliquant l'extension de cet outil (fonctions de lyapunov *non monotones*) [51] pour essayer éventuellement de rendre cet outil moins conservatif.

ANNEXES

A.1. Les cycles limites pour une PLL analogique non linéaire

%FONCTION 1: PLLdyna

function dx = PLLdyna(t,x)

global P1 P3

dx = [x(2);-P1*x(2)-cos(x(1))+P3];

%FONCTION 2: PLLbifurca (LES CYCLES LIMITES POUR UNE PLL ANALOGIQUE NON LINÉAIRE)

clear all

global P1 P3

% Variables permettant d'effectuer un zoom sur un point d'attraction

P1=0.5;

P3=0.7;

np=30;

xmin=-2*pi; xmax=4*pi+xmin; dx=(xmax-xmin)/np;

ymin=-4; ymax=4; dy=(ymax-ymin)/np;

figure(1)

% Pour afficher les positions et vitesses sous forme de flèches aux points (x,y)

[x,y] = meshgrid(xmin:dx:xmax,ymin:dy:ymax);

u=y; v= -P1*y-cos(x)+P3;

quiver(x,y,u,v)

title(['P1=' num2str(P1),' P3=' num2str(P3)],'fontsize',18)

hold on

for ii=0:np

[tt,ty]=ode15s(@PLLdyna,[0 20],[xmin ymin+ii*dy]);

plot(ty(:,1),ty(:,2),'b','linewidth',1)

[tt,ty]=ode15s(@PLLdyna,[0 20],[xmax ymin+ii*dy]);

plot(ty(:,1),ty(:,2),'b','linewidth',1)

[tt,ty]=ode15s(@PLLdyna,[0 20],[xmin+ii*dx ymin]);

plot(ty(:,1),ty(:,2),'b','linewidth',1)

[tt,ty]=ode15s(@PLLdyna,[0 20],[xmin+ii*dx ymax]);

plot(ty(:,1),ty(:,2),'b','linewidth',1)

end

hold off

axis equal

```
axis([xmin,xmax,ymin,ymax])  
grid on
```


A.2. Stabilité des réseaux autonomes d'ADPLLs idéaux

Un réseau autonome d'ADPLLs est constitué de N nœuds, tous les nœuds étant caractérisés par les mêmes coefficients K_1 et K_2 (grandeurs numériques) mais pas nécessairement les mêmes périodes propres (grandeurs analogiques). Chacun des nœuds « voit » un certain nombre de voisins, de sorte qu'on définit *l'erreur de phase totale* au k^{ieme} nœud $e_k[n]$ par :

$$e_k[n] = \frac{1}{|V_k|} \sum_{l \in V_k} e_{lk}[n], \quad (\text{A.1.1})$$

où V_k est l'ensemble des indices des nœuds connectés au nœud k et $|V_k|$ le cardinal de V_k . L'erreur temporelle entre le nœud k et le nœud l est $e_{lk}[n]$. Notons que :

$$e_{lk}[n] = -e_{kl}[n], \quad (\text{A.1.2})$$

et on définit la *matrice d'adjacence* \mathbf{A} comme la matrice de terme générique α_{ij} [91-92, 96, 99]. Cette matrice n'est pas nécessairement symétrique. On définit également le *Laplacien* \mathbf{L} du réseau par :

$$\mathbf{L} = \mathbf{I}_N - \mathbf{A} \quad (\text{A.1.3})$$

Avec ces définitions, on peut écrire les équations régissant les erreurs de phase totales de chaque nœud. On obtient la relation suivante :

$$\mathbf{e}[n+1] - 2\mathbf{e}[n] + \mathbf{e}[n-1] = -(K_1 \mathbf{L} \mathbf{e}[n] + K_2 \mathbf{L} \mathbf{e}[n-1]) \quad (\text{A.1.5})$$

où $\mathbf{e}[n]$ est le vecteur dont la $i^{\text{ème}}$ coordonnée est $e_i[n]$. L'équation (A.1.5) peut encore se mettre sous la forme :

$$\mathbf{x}[n+1] = \mathbf{\Gamma} \mathbf{x}[n] \quad (\text{A.1.6})$$

où

$$\mathbf{x}[n] = \begin{pmatrix} \mathbf{e}[n] \\ \mathbf{e}[n-1] \end{pmatrix} \quad (\text{A.1.7})$$

et

$$\mathbf{\Gamma} = \begin{pmatrix} 2\mathbf{I}_N - K_1\mathbf{L} & -\mathbf{I}_N - K_2\mathbf{L} \\ \mathbf{I}_N & \mathbf{0}_N \end{pmatrix} \quad (\text{A.1.8})$$

La stabilité de (A.1.6) (et donc la convergence du réseau vers un état synchrone) est fonction des valeurs propres de $\mathbf{\Gamma}$: si celles-ci sont toutes situées à l'intérieur du cercle unité, le réseau est stable.

Remarquons que, en vertu de (A.1.3), la somme de chaque ligne du Laplacien du réseau vaut 0. En conséquence, la somme de chaque ligne de $\mathbf{\Gamma}$ vaut 1, quelles que soient les valeurs de K_1 et K_2 . Ainsi, tout vecteur de la forme

$$\mathbf{x} = \lambda \mathbf{1}_{2N} \quad (\text{A.1.9})$$

est un vecteur propre de $\mathbf{\Gamma}$, associé à la valeur propre 1. Ceci signifie que (A.1.6) a un état d'équilibre (stable ou instable selon la valeur des autres valeurs propres de $\mathbf{\Gamma}$) dans lequel toutes les erreurs de phase totales sont égales. Du fait de la relation (A.1.2), on peut montrer que le seul état d'équilibre possible est celui correspondant à $\lambda = 0$, i.e. à des erreurs de phase totales identiquement nulles.

Enfin, soulignons le fait que les fronts montants du réseau $\mathbf{t}[n]$ sont régis par la même équation que l'erreur de phase totale (équation (A.1.5)). En conséquence, la stabilité et la convergence en phase sont équivalentes à la stabilité et à la convergence en fréquence.

A.2.1. Réseaux complètement connectés

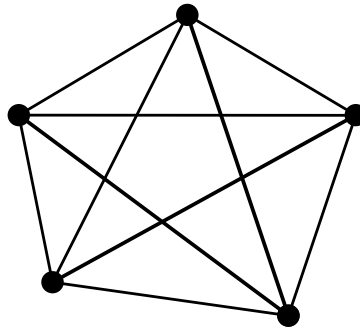


Fig. A.1.1 - Réseau complètement connecté de 5 nœuds.

Un réseau complètement connecté est un réseau où chaque nœud « voit » tous les autres (Fig. A.1.1). Pour un réseau de taille N , on impose ici :

$$\alpha_{ij} = \frac{1}{N-1}, \quad \forall i \neq j. \quad (\text{A.1.10})$$

On a représenté aux Fig. A.1.2, A.1.3 et A.1.4 les zones de stabilité de réseaux complètement connectés de taille 2, 4 et 16. On constate que le domaine de stabilité va en grandissant avec la taille du réseau. Pour les grandes tailles, on semble converger vers le même domaine de stabilité que pour une SS-ADPLL idéale seule (délimité par les traits bleus et l'axe horizontal).

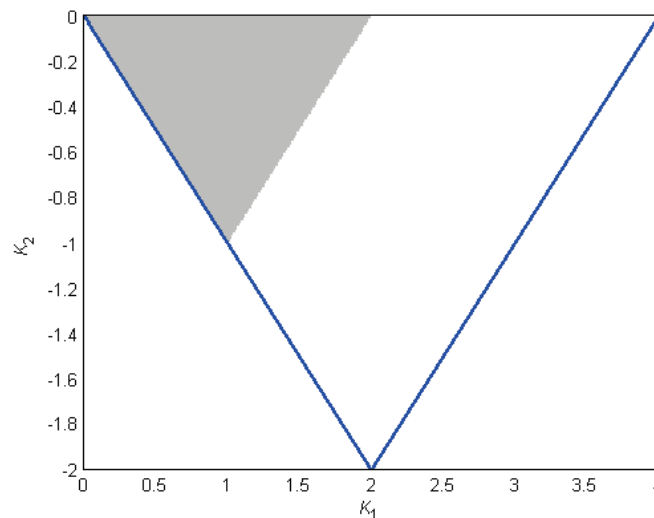


Fig. A.1.2 - Domaine de stabilité (en grisé) d'un réseau complètement connecté pour $N = 2$.

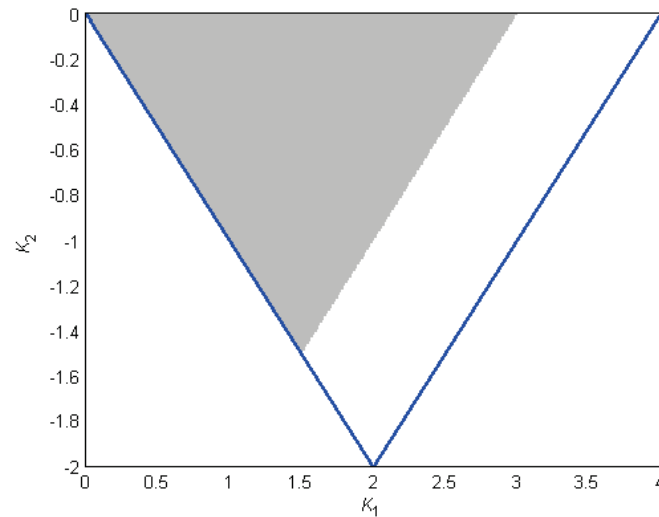


Fig. A.1.3 - Domaine de stabilité (en grisé) d'un réseau complètement connecté pour $N = 4$.

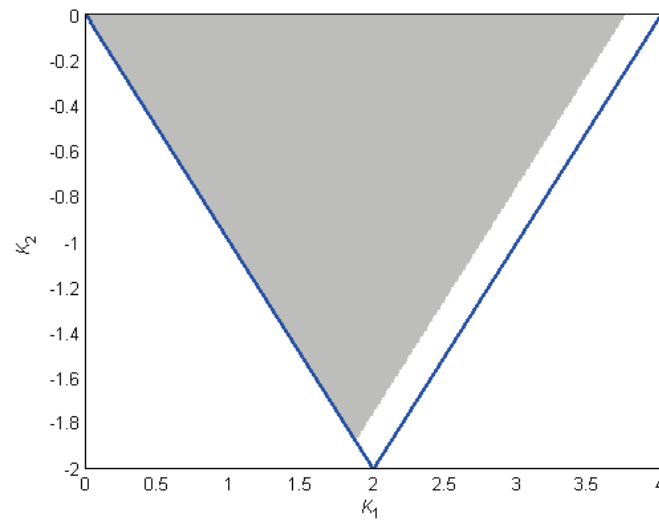


Fig. A.1.4 - Domaine de stabilité (en grisé) d'un réseau complètement connecté pour $N = 16$.

On voit à travers ces figures, comment la taille de réseaux complètement connectés influe sur leurs domaines de stabilité.

A.2.2. Réseaux cartésiens

Un réseau cartésien est représenté à la Fig. A.1.5. Dans un réseau cartésien, chaque nœud du réseau n'est relié qu'à ses plus proches voisins. Les coefficients de pondération sont définis comme suit :

$$\alpha_{ij} = \frac{1}{c \cdot r(\mathfrak{C}_i)} \quad (\text{A.1.11})$$

Notons que la matrice d'adjacence n'est plus symétrique : par exemple $\alpha_{12} = 1/2$ et $\alpha_{21} = 1/3$, $\alpha_{25} = 1/3$ et $\alpha_{52} = 1/4$.

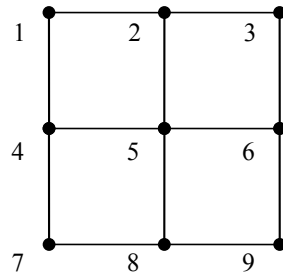


Fig. A.1.5 - Réseau cartésien de 9 nœuds.

Le domaine de stabilité d'un réseau cartésien n'est pas affecté par la taille du réseau. On trouve, pour tout N et quel que soit le nombre de colonnes et de lignes constituant le réseau, le domaine de stabilité représenté à la Fig. A.1.6.

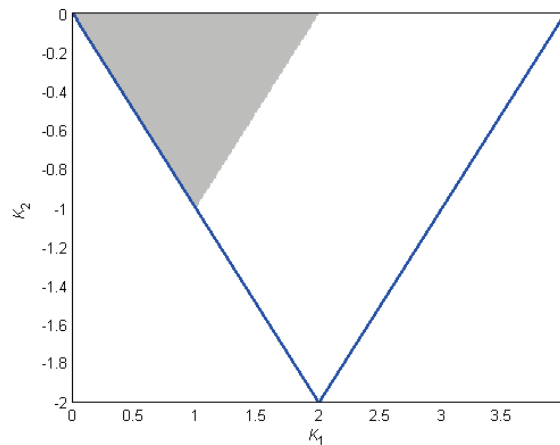


Fig. A.1.6 - Domaine de stabilité (en grisé) d'un réseau quadratique de taille quelconque.

Notons que la position des valeurs propres de Γ n'est globalement pas affectée par la taille du réseau. On a représenté à la Fig. A.1.7 les valeurs propres de Γ pour un réseau de 9 nœuds et pour un réseau de 36 nœuds, pour un correcteur défini par $K_1 = 1$ et $K_2 = -0.6$. Si les valeurs propres des deux réseaux semblent être réparties de la même manière à l'intérieur du cercle unité, celles correspondant à $N = 36$ sont plus proches de celui-ci que celles correspondant à $N = 9$. En conséquence, un réseau cartésien de grande dimension aura sans doute une marge de stabilité plus faible qu'un réseau cartésien de grande dimension.

Le même phénomène peut être observé pour un réseau complètement connecté (Fig. A.1.8) à la différence que, pour ce type de réseau, les valeurs propres non-triviales (différentes de 1) ont une multiplicité de $N-1$ et que, lorsque la taille augmente, la position de ces valeurs propres ne semble que peu affectée. La marge de stabilité d'un tel réseau n'est donc que modérément affectée par l'extension du nombre de nœuds.

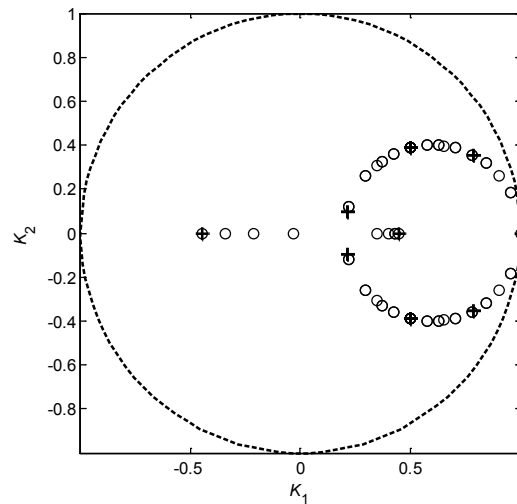


Fig. A.1.7 - Répartition des valeurs propres de Γ à l'intérieur du cercle unité pour un réseau quadratique de taille $N = 9$ (croix) ou $N = 36$ (ronds).

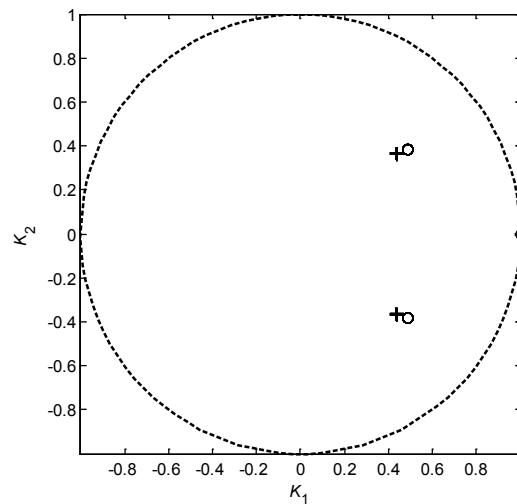


Fig. A.1.8 - Répartition des valeurs propres de Γ à l'intérieur du cercle unité pour un réseau complètement connecté de taille $N = 9$ (croix) ou $N = 36$ (ronds). Pour un réseau de taille beaucoup plus importante, les valeurs propres non-triviales de Γ sont très proches des ronds.

A.3. Simulation de réseaux de SS-ADPLLs de type A en présence de non-idéalités

Les domaines de stabilité pour des réseaux de SS-ADPLLs de type A sont donnés de la Fig. A.2.1 à la Fig. A.2.5. Les résultats obtenus valident la condition nécessaire de stabilité issue de l'équation maitresse obtenue préalablement en section 4.2.2 pour les SS-ADPLLs de type A. Les simulations mettent aussi en évidence que le domaine de stabilité de ces types de réseaux diminue avec la taille du réseau tout comme pour leurs modèles idéalisés (Chapitre 4).

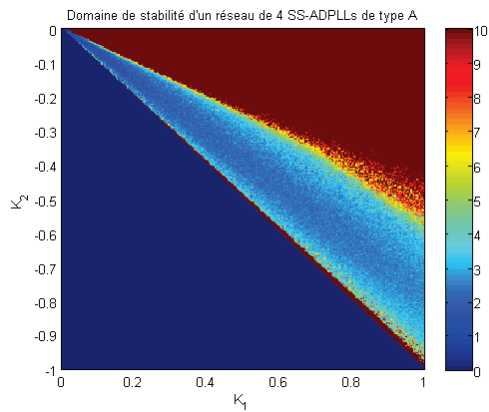


Fig. A.2.1 Domaine de stabilité d'un réseau de 4 SS-ADPLLs passésistes de type A

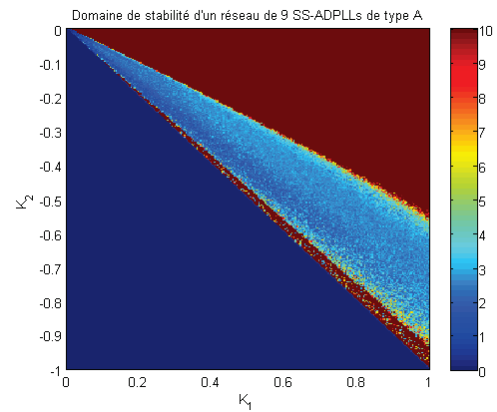


Fig. A.2.2 Domaine de stabilité d'un réseau de 9 SS-ADPLLs passésistes de type A

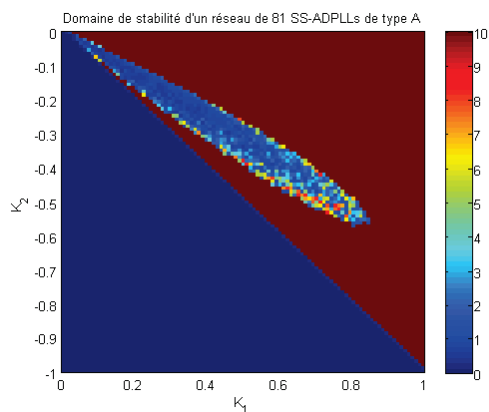


Fig. A.2.3 Domaine de stabilité d'un réseau de 81 SS-ADPLLs passésistes de type A

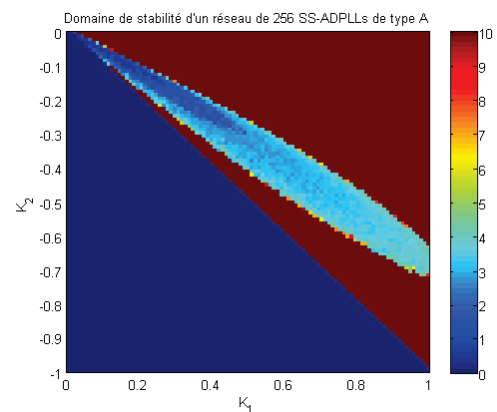


Fig. A.2.4 Domaine de stabilité d'un réseau de 256 SS-ADPLLs passésistes de type A

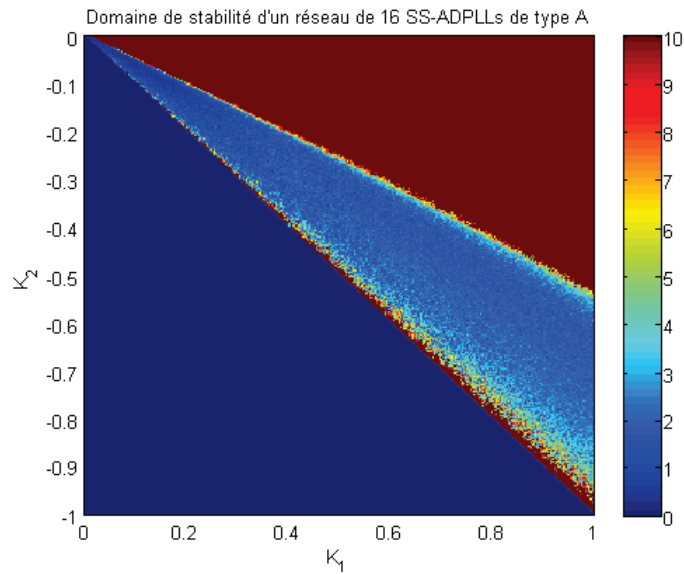


Fig. A.2.5 Domaine de stabilité d'un réseau de 16 SS-ADPLLs passésistes de type A

De même que pour les réseaux de SS-ADPLLs de type B, on constate que le lieu des coefficients des filtres permettant de maintenir le réseau stable (lorsque la taille du réseau augmente) va dans le même sens que ceux du choix des coefficients optimums préconisés par la méthode prenant la mise en forme bruit de quantification (Section 4.4): coefficients de filtre proches de zéro.

A.4. Génération des matrices d'adjacence

```
function A=generate_connectivity_mtx(Nl,Nc,type)
%A=generate_connectivity_mtx(Nl,Nc,type)
%Génère la matrice d'adjacence d'un réseau de PLLs
%Nl = nb de lignes
%Nc = nb de colonnes
%type = 'quad' pour un réseau quadratique, 'connected' pour un réseau
%complètement connecté

A=zeros(Nl*Nc);
if strcmp(type,'quad')

    for k=1:Nl,
        for l=1:Nc,
            Node_nbr=l+(k-1)*Nc;
            if l==1 %on est sur la colonne de gauche

                if k==1 %on est sur le coin sup gauche
                    A(Node_nbr,Node_nbr)=0;
                    A(Node_nbr,Node_nbr+1)=1/2;
                    A(Node_nbr,Node_nbr+Nc)=1/2;
                elseif k==Nl %on est sur le coin inf gauche
                    A(Node_nbr,Node_nbr)=0;
                    A(Node_nbr,Node_nbr+1)=1/2;
                    A(Node_nbr,Node_nbr-Nc)=1/2;
                else % on n'est pas sur un coin
                    A(Node_nbr,Node_nbr)=0;
                    A(Node_nbr,Node_nbr+1)=1/3;
                    A(Node_nbr,Node_nbr-Nc)=1/3;
```

```

    A(Node_nbr,Node_nbr+Nc)=1/3;
end

```

```

elseif l==Nc % on est sur la colonne de droite

```

```

    if k==1 %on est sur le coin sup droit
        A(Node_nbr,Node_nbr)=0;
        A(Node_nbr,Node_nbr-1)=1/2;
        A(Node_nbr,Node_nbr+Nc)=1/2;
    elseif k==Nl %on est sur le coin inf droit
        A(Node_nbr,Node_nbr)=0;
        A(Node_nbr,Node_nbr-1)=1/2;
        A(Node_nbr,Node_nbr-Nc)=1/2;
    else % on n'est pas sur un coin
        A(Node_nbr,Node_nbr)=0;
        A(Node_nbr,Node_nbr-1)=1/3;
        A(Node_nbr,Node_nbr-Nc)=1/3;
        A(Node_nbr,Node_nbr+Nc)=1/3;
    end

```

```

else

```

```

    if k==1 %on est sur la ligne du haut
        A(Node_nbr,Node_nbr)=0;
        A(Node_nbr,Node_nbr-1)=1/3;
        A(Node_nbr,Node_nbr+1)=1/3;
        A(Node_nbr,Node_nbr+Nc)=1/3;
    elseif k==Nl %on est sur la ligne du bas
        A(Node_nbr,Node_nbr)=0;
        A(Node_nbr,Node_nbr-1)=1/3;
        A(Node_nbr,Node_nbr+1)=1/3;
        A(Node_nbr,Node_nbr-Nc)=1/3;
    end

```

```

else
    A(Node_nbr,Node_nbr)=0;
    A(Node_nbr,Node_nbr-1)=1/4;
    A(Node_nbr,Node_nbr+1)=1/4;
    A(Node_nbr,Node_nbr-Nc)=1/4;
    A(Node_nbr,Node_nbr+Nc)=1/4;
end
end
end
end
elseif strcmp(type,'connected')
    N=Nl*Nc;
    A=1/(N-1)*ones(N)-1/(N-1)*eye(N);
else
    disp('Type de réseau non reconnu')
end

```

A.5. Simulation d'un réseau de SS-ADPLLs

%Ce script remplit une matrice S contenant 1 si le réseau est "stable", 0 sinon

N_edge=1e3;%Nb de fronts montants maximum

N_x=3;%Nb de lignes

N_y=3;%Nb de colonnes

N_nodes=N_x*N_y;

A=generate_connectivity_mtx(N_x,N_y,'quad');

L=eye(N_nodes)-A;%matrice laplacienne

K1_vec=linspace(0,4,100);%Valeurs de K1 balayées

K2_vec=linspace(2,0,100);%Valeurs de K2 balayées

S=zeros(length(K1_vec),length(K2_vec));

dt_stock=zeros(N_nodes,N_edge);%Stockage des périodes des VCO

E_stock=zeros(N_nodes,N_edge);%Stockage des erreurs de phase totales

for p=1:length(K1_vec),

 K1=K1_vec(p);

 for q=1:length(K2_vec),

 K2=K2_vec(q);

 t_n_moins_1=0.1*rand(N_nodes,1);%Initialisation des horloges

 T0=1+0*randn(N_nodes,1);%Initialisation des periodes des VCO

 Y=zeros(N_nodes,1);%Initialisation des commandes des VCO

 epsilon_n=zeros(N_nodes,1);%Initialisation des erreurs des DPD

 t_n=t_n_moins_1+T0;

 T_old_j=t_n_moins_1(:,ones(1,N_nodes));

```

T_old_ij=T_old_j'-T_old_j;%Definition des erreurs de phase entre tous les noeuds

while k<N_edge && ~any(DT)>5 && ~any(DT)<0,
    T_n_j=t_n(:,ones(1,N_nodes));
    T_n_ij=T_n_j'-T_n_j;%Definition des erreurs de phase entre tous les noeuds
    M=T_n_ij>=0;%Definition des signes des erreurs de phase entre tous les noeuds

    %e_n_moins_1=epsilon_n;% si propagation d'erreur
    epsilon_n=sum(A.*(M.*T_old_ij+(1-M).*T_n_ij),2);% si passéiste
    epsilon_n=sum(A.*((1-M).*T_n_ij),2);% si nihiliste
    e_n_moins_1=sum(A.*T_old_ij,2);% si pas de propagation d'erreur

    Y=Y+K1*epsilon_n+K2*e_n_moins_1;% actualisation de la commande des VCO
    DT=T0+Y;
    t_n=t_n+DT;% actualisation des fronts montants
    T_old_ij=T_n_ij;
    dt_stock(:,k)=DT;% stockage de la periode du VCO
    E_stock(:,k)=L*t_n;% stockage de l'erreur de phase totale
    k=k+1;
end
if any(DT)>5 || any(DT)<0 || any(isnan(DT)) || any(isnan(e_n_moins_1)),
    S(p,q)=0;
else
    S(p,q)=1;
    figure(121)
    plot(dt_stock');
    figure(122)
    plot(E_stock')
end
end
end
figure(123)
pcolor(K1_vec,K2_vec,S');

```

```
colormap(1-0.25*gray);  
shading flat
```

A.6. Simulation de réseaux de SS-ADPLLs (prise en compte de la quantification et de la saturation)

```
%Ce script remplit une matrice S contenant 1 si le réseau est "stable", 0 sinon
%Si les vecteurs K1_vec et K2_vec n'ont qu'un élément, il trace la valeur
%de la période des VCO au cours du temps
%Le DPD correspond au DPD d'Eldar avec des niveaux réglables de
%quantification (Quantif) et de saturation (Satur)
%On peut décider de mettre une ou des horloges de référence (Ref)
%On a le choix de l'architecture de filtre (passéiste/nihiliste/avec
%ou sans propagation d'erreur) en commentant les lignes appropriées
N_vec=[4];
for N_exp=1:length(N_vec),

    N_edge=3000;%Nb de fronts montants maximum
    N_x=N_vec(N_exp);%Nb de lignes
    N_y=N_x;%Nb de colonnes

    Ref=[1];%Numero de l'horloge de reference (il peut y en avoir plusieurs ou aucune
    (dans ce cas laisser Ref = []))
    Sigma_bruit=0.00;
    Quantif = 8e-3;%0.12/15;%Niveau de quantification du DPD
    Satur = 16;%150;%Niveau de la saturation du DPD

    N_nodes=N_x*N_y;
    A=generate_connectivity_mtx(N_x,N_y,'quad');
    %N_nodes=1;A=0;%PLL seule
    %A=[0 1;1 0];%reseau de deux
    %A=[0 1/2 1/2;1/2 0 1/2;1/2 1/2 0];%reseau de trois
    %A=[0 1/2 0 1/2;1/2 0 1/2 0; 0 1/2 0 1/2;1/2 0 1/2 0];
    %A=[0 1/3 0 1/3 1/3;1/3 0 1/3 0 1/3;0 1/3 0 1/3 1/3;1/3 0 1/3 0 1/3;1/4 1/4 1/4
    1/4 0];%reseau de cinq (carre centre)
    %A=[0 1 0;1/2 0 1/2;0 1 0];
    % N_nodes=5;
    % A=[0 1/3 1/3 0 1/3;1/2 0 0 1/2 0;1/2 0 0 1/2 0;0 1/2 1/2 0 0;1 0 0 0 0];
    L=eye(N_nodes)-A;%matrice laplacienne

    %%%%%%%%%NEW TAB%%%%%%%%%%%%%%
    v=zeros(1,N_nodes);
    for i=1:N_nodes,
        v(i)=(-1)^(length(find(L(i,:)))-1);
    end
    %%%%%%%%%%%%%%%

%    K1_vec=linspace(0,1,300);%Valeurs de K1 balayées
```

```

% K2_vec=linspace(-1,0,300);%Valeurs de K2 balayées
% K1_vec=linspace(0.5,0.5,1);%Valeurs de K1 balayées8
% K2_vec=linspace(-0.3,-0.3,1);%Valeurs de K2 balayées
% K1_vec=linspace(0.2,0.2,1);%Valeurs de K1 balayées
% K2_vec=linspace(-0.19,-0.19,1);%Valeurs de K2 balayées

K1_vec=linspace(0.08,0.08,1);%Valeurs de K1 balayées
K2_vec=linspace(-0.0797,-0.0797,1);%Valeurs de K2 balayées

S=ones(length(K1_vec),length(K2_vec));R=S;
dt_stock=zeros(N_nodes,N_edge-1);%Stockage des périodes des VCO
E_stock=zeros(N_nodes,N_edge-1);%Stockage des erreurs de phase totales
t_n_start=Quantif*rand(N_nodes,1);
t_n_start(Ref)=0;
T0=ones(N_nodes,1)+0*Quantif*rand(N_nodes,1);%Initialisation des periodes des
VCO
T0(Ref)=1;
T0_min=T0-0.02-0.1*rand(size(T0));
T0_max=T0+0.02+0.1*rand(size(T0));

for p=1:length(K1_vec),
    K1=K1_vec(p);
    for q=1:length(K2_vec),
        K2=K2_vec(q);

        if K2>=-K1,% NOUVELLE CONDITION POUR REGARDER SEULEMENT DANS D

            t_n_moins_1=t_n_start;%Initialisation des horloges
            Y=zeros(N_nodes,1);%Initialisation des commandes des VCO
            epsilon_n=zeros(N_nodes,1);%Initialisation des erreurs des DPD
            e_n_moins_1=zeros(N_nodes,1);

            t_n=t_n_moins_1+T0;
            T_old_j=t_n_moins_1(:,ones(1,N_nodes));
            %T_old_ij=T_old_j'-T_old_j;
            T_old_ij=quantify_error(T_old_j'-T_old_j,Quantif,Satur);%Definition des erreurs
de phase entre tous les noeuds
            DT=zeros(1,N_nodes);k=1;

            while k<N_edge,
                t_n=t_n+0.000*randn(size(t_n));
                T_n_j=t_n(:,ones(1,N_nodes));
                %T_n_ij=T_n_j'-T_n_j;
                T_n_ij=quantify_error(T_n_j'-T_n_j,Quantif,Satur);%Definition des erreurs de
phase entre tous les noeuds
                M=T_n_ij>0;%Definition des signes des erreurs de phase entre tous les noeuds

                %            e_n_moins_2=e_n_moins_1;% si propagation d'erreur

```



```

e_n_moins_1=epsilon_n;% si propagation d'erreur
epsilon_n=sum(A.*(M.*T_old_ij+(1-M).*T_n_ij),2);% si passéiste
%epsilon_n=sum(A.*T_n_ij,2);%si ideale
%epsilon_n=sum(A.*((1-M).*T_n_ij),2);% si nihiliste
%e_n_moins_1=sum(A.*T_old_ij,2);% si pas de propagation d'erreur

Y=Y+K1*epsilon_n+K2*e_n_moins_1;% actualisation de la commande des VCO
%Y=Y+K1/2*epsilon_n+(K2/2+K1/2)*e_n_moins_1+K2/2*e_n_moins_2;%
actualisation de la commande des VCO pour le système moyen
%Y=Y+K1/2*epsilon_n+(K2+K1/2)*e_n_moins_1;% actualisation de la
commande des VCO pour le système moyen
Y(Ref)=0;
DT=min(max(T0+Y,T0_min),T0_max);
E_stock(:,k)=L*t_n;% stockage de l'erreur de phase totale
t_n=t_n+DT;% actualisation des fronts montants
dt_stock(:,k)=DT;% stockage de la periode du VCO
T_old_ij=T_n_ij;
m_t_n=mean(t_n);
k=k+1;
%      if k>6000 && any(abs(e_n_moins_1)>4*Quantif),
%          S(p,q)=0;
%          break
%      end
end
E_master=zeros(1,100);
for ii=1:100,
    E_master(ii)=v*E_stock(:,end-100+ii);
end
S(p,q)=abs(mean(E_master))/Quantif;
%R(p,q)=mean(mean(abs(E_stock(:,end-100+ii))))/Quantif;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%JM%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
E_stockr=zeros(1,100);
for ii=1:100,
    E_stockr(ii)=mean(abs(E_stock(:,end-100+ii)));
end
R(p,q)=mean(E_stockr)/Quantif;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%JM%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%      if norm(e_n_moins_1)>1,
%          S(p,q)=S(p,q)*2;
%      end

```

```

else
    S(p,q)=0;
    R(p,q)=0;
end
% if any((DT)>5) || any(DT<0) || any(isnan(DT)) || any(isnan(e_n_moins_1))
|| any(E_stock(:,end-100)>10*Quantif),
%     S(p,q)=0;
%     break
% end

end
end
if length(K1_vec)>1 && length(K2_vec)>1
    figure%(1029+N_exp)
    %subplot(1,2,1)
    pcolor(K1_vec,K2_vec,S');caxis([0 10])
    title(['Erreur "maître" pour ', num2str(N_vec(N_exp)),' noeuds'])
    shading flat;colorbar
    figure
    %subplot(1,2,2)
    pcolor(K1_vec,K2_vec,R');caxis([0 10])
    title(['Erreur totale pour ', num2str(N_vec(N_exp)),' noeuds'])
    shading flat;colorbar

else
    figure(3000+N_exp);
    plot(cumsum(dt_stock'))
    title('Periodes')
    figure(2000+N_exp)
    stem(cumsum(dt_stock(:,end-100:end))+[1;zeros(100,1)]*sum(dt_stock(:,1:end-101)))-min(sum(dt_stock(:,1:end-101))),ones(size(cumsum(dt_stock(:,end-100:end)))));
    title('Fronts')
    figure(1000+N_exp)
    plot(E_stock');
    title('Erreurs totales')
    %hold on
    figure
    E_master=zeros(1,length(E_stock));
    for ii=1:length(E_stock),
        E_master(ii)=v*E_stock(:,ii);
    end
    plot(E_master,'r','LineWidth',2)
    title('Master')
    %hold off
end
%save toto S K1_vec K2_vec
end

```

A.7. Détermination de Fonctions de Lyapunov pour des systèmes linéaires par morceaux

```

clear all;close all;clc
K1_vec=linspace(0,4,300);
K2_vec=linspace(-2,0,300);
Souv=ones(length(K1_vec),length(K2_vec));
for p=1:length(K1_vec)
    K1=K1_vec(p);
    for q=1:length(K2_vec)
        K2=K2_vec(q);
        % PREDICTION PASSEISTE
        A1=[2-K1 -(1+K2) 0;1 0 0;0 1 0];
        A2=[2 -(1+K1+K2) 0;1 0 0;0 1 0];
        A3=[2 -(1+K1) -K2;1 0 0;0 1 0];
        A4=[2-K1 -1 -K2;1 0 0;0 1 0];

        E1=[-1 0 0;0 0 0;0 0 0];
        E2=[1 0 0;0 0 0;0 0 0];
        E3=[1 0 0;0 0 0;0 0 0];
        E4=[-1 0 0;0 0 0;0 0 0];
        n=size(A1,1)
        setlmis([]) % initialisation du systeme de LMI

        P1=sdpvar(n); P2=sdpvar(n); P3=sdpvar(n); P4=sdpvar(n);

U1=sdpvar(n,n,'full');U2=sdpvar(n,n,'full');U3=sdpvar(n,n,'full');U4=sdpvar(n,n,'full');
        W1=sdpvar(n,n,'full');W3=sdpvar(n,n,'full');
        Q12=sdpvar(n,n,'full');
        Q23=sdpvar(n,n,'full');Q24=sdpvar(n,n,'full');
        Q34=sdpvar(n,n,'full');
        Q41=sdpvar(n,n,'full');Q42=sdpvar(n,n,'full');

        F = SET([])

F=[F,U1>0,U2>0,U3>0,U4>0,W1>0,W3>0,Q12>0,Q23>0,Q24>0,Q34>0,Q41>0,Q42>0];
        %MATRICES DEFINIES POSITIVES
        F=[F,P1-E1'*U1*E1>0,P2-E2'*U2*E2>0,P3-E3'*U3*E3>0,P4-E4'*U4*E4>0];
        %STABILITE DANS LES CELLULES
        F=[F,A1'*P1*A1-P1+E1'*W1*E1<0,A3'*P3*A3-P3+E3'*W3*E3<0];
        %TRANSITIONS POSSIBLES
        F=[F,A1'*P2*A1-P1+E1'*Q12*E1<0];
        F=[F,A2'*P3*A2-P2+E2'*Q23*E2<0,A2'*P2*A2-P2+E2'*Q24*E2<0];

```

```

F=[F,A2'*P1*A2-P2+E2'*Q34*E2<0];
F=[F,A4'*P1*A1-P4+E1'*Q41*E1<0,A2'*P1*A2-P2+E2'*Q42*E2<0];
% F=[F;P1==P2];

solution = solvesdp(F)

v1=eig(double(P1));
v2=eig(double(P2));
v3=eig(double(P3));
v4=eig(double(P4));

%if v1 >0 & v2 >0,
  if v1 >0 & v3 >0,
    if (double(U1))>0 & (double(U3))>0,
      if (double(W1))>0 & (double(W3))>0,
        if (double(Q12)) & (double(Q34))>0,
          Souv(p,q)=0;
        end
      end
    end
  end
end

end
end
pcolor(K1_vec,K2_vec,Souv');colormap([0.75 0.75 0.75;1 1 1]);shading flat

```

LISTE DES PUBLICATIONS

Article de revue avec comité de lecture

J-M. N. Akré, J. Juillard, D. Galayko, and E. Colinet “*Synchronization Analysis of Networks of Self-Sampled All-Digital Phase-Locked Loops*”, IEEE Transactions on Circuits and Systems I, Vol.59, pp. 708-720, April 2012.

Conférences Internationales avec Actes et Comité de Lecture

[1] **J-M. N. Akré**, J. Juillard, D. Galayko, and E. Colinet, “*Synchronized State in Networks of Digital Phase-Locked Loops*”, 8th IEEE International NEWCAS Conference, pp. 89-92, Montreal, Canada, 2010.

[2] **J-M. N. Akré**, J. Juillard, S. Olaru, D. Galayko, and E. Colinet, “*Determination of the Behaviour of Self-Sampled Digital Phase-Locked Loops*”, 53rd IEEE International MWSCAS’10, pp. 1089-1092, Seattle, Washington (USA), August 1-4, 2010.

[3] **J-M. N. Akré**, J. Juillard, M. Javidan, E. Zianbetov, D. Galayko, A. Korniienko, and E. Colinet, “*A Design Approach for Networks of Self-Sampled All-Digital Phase-Locked Loops*”, 20th European Conference on Circuit Theory and Design (ECCTD), Linköping, Sweden, pp. 725-728, 2011.

[4] M. Javidan, E. Zianbetov, F. Anceau, D. Galayko, A. Korniienko, E. Colinet, **J-M. N. Akré**, J. Juillard, “*All-digital PLL array provides reliable distributed clock for SOC’s*”, 2011 IEEE International Symposium on Circuits and Systems (ISCAS), Rio de Janeiro, Brazil, May 15-18, pp. 2589-2592, 2011.

RÉFÉRENCES BIBLIOGRAPHIQUES

- [1] G.A. Pratt, J. Nguyen, "*Distributed Synchronous Clocking*", IEEE Transactions on Parallel and Distributed System, vol. 6, pp. 314-328, 1995.
- [2] E.G. Friedman, "*Clock Distribution Networks in Synchronous Digital Integrated Circuits*", Proceedings of the IEEE, vol. 89, pp. 665-692, 2001.
- [3] V. Gutnik, A.P. Chandrakasan, "*Active GHz Clock Network Using Distributed PLLs*", IEEE Journal of Solid-State Circuits, vol. 35, pp. 1553-1560, 2000.
- [4] R. Olfati-saber, J.A. Fax, and R.M. Murray "*Consensus and Cooperation in Networked Multi-Agent Systems*", Proceedings of the IEEE, vol. 95, pp. 215-233, 2007.
- [5] W. Ren, et al., "*A Survey of Consensus Problems in Multi-agent Coordination*", American Control Conference, pp. 1859-1864, 2005.
- [6] H.Y. Hsieh, et al., "*Self-calibrating Clock Distribution with Scheduled Skews*", Proceedings of the 1998 IEEE international symposium on circuits and systems, vol. 2, pp. 470-473, 1998.
- [7] H.Sutoh and K. Yamakoshi, "*A clock distribution technique with an automatic skew compensation circuit*", c. Electron, pp. 277-283, 1998.
- [8] A.M Bueno, A.A. Ferreira, J.R.C. Piqueira, "*Modeling and filtering double-frequency jitter in one-way master-slave chain networks*", IEEE Transactions on Circuits and Systems I, vol. 57, pp. 3104-3111, 2010.
- [9] F. Anceau, "*Une technique de réduction de la puissance dissipée par l'horlogerie des circuits complexes rapides*", 4ème journées francophones d'étude Faible Tension Faible Consommation (FTFC'2003), Paris, 16 mai 2003.
- [10] T. Raghebn et al., "*Design of thermally robust clock trees using dynamically adaptive clock buffers*", IEEE Transactions on Circuits and Systems I, vol. 56, pp. 374-383, 2009.
- [11] A. Rajaram, D.Z. Pan, "*Robust chip-level clock tree synthesis*", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 30, pp. 877-890, 2011.
- [12] E.G. Friedman, "*Design and Analysis of a Hierarchical Clock Distribution System for Synchronous Standard Cell/Macrocell VLSI*", IEEE Journal of Solid-State Circuits, vol. 21, pp. 240-246, 1986.

- [13] M. Saint-Laurent, M. Swaminathan, " *A Multi-PLL Clock Distribution Architecture for Gigascale Integration*", IEEE Computer Society Workshop on VLSI, pp. 30-35, 2001.
- [14] S.H. Strogatz, " *Sync: The Emerging Science of Spontaneous Order*", Hyperion, New York, 2003.
- [15] W.C. Lindsey et al., " *Network Synchronization*," Proceedings of the IEEE, vol. 73, no.10, Oct. 1985, pp. 1445–1467.
- [16] O. Simeone et al., " *Distributed synchronization in wireless networks*," Proc. IEEE, vol. 25, Sept. 2008, pp. 81–97.
- [17] J-M. Akaré, J. Juillard, D. Galayko, and E. Colinet " *Synchronized State in Networks of Digital Phase-Locked Loops*", 8th IEEE International NEWCAS Conference, pp. 89-92, Montreal, 2010.
- [18] J-M. Akaré, J. Juillard, S. Olaru, D. Galayko, and E. Colinet " *Determination of the Behaviour of Self-Sampled Digital Phase-Locked Loops*", 53rd IEEE International MWSCAS'10, pp. 1089-1092, Seattle, Washington (USA), August 1-4, 2010.
- [19] R. Flynn, and O. Feely, " *Limit Cycles in Digital Bang-Bang Phase-Locked Loops*", 18th European Conference on Circuit Theory and Design (ECCTD), Aug. 2007, pp 731-734.
- [20] R.C. Walker, " *Designing Bang-Bang PLLs for Clock and Data Recovery in Serial Data Transmission Systems*", Phase-Locking in High-Performance Systems, B. Razavi, Ed.:IEEE Press, pp 34-45, 2003.
- [21] W. Namgoong, " *Observer-controller digital PLL*", IEEE Transactions on Circuits and Systems I, vol. 57, pp. 631-641, 2010.
- [22] M. Zanuso, D. Tasca, S. Levantino, A. Donadel, C. Samori, A. Lacaita, " *Noise analysis and minimization in bang-bang digital PLLs*", IEEE Transactions on Circuits and Systems II, vol. 56, pp. 835-839, 2009.
- [23] M. Kumm, H. Klingbeil, P. Zipf, " *An FPGA-based linear all-digital phase-locked loop*", IEEE Transactions on Circuits and Systems I, vol. 57, pp. 2487-2497, 2010.
- [24] N. Da Dalt, " *A design-oriented study of the nonlinear dynamics of digital bang-bang PLLs*", IEEE Transactions on Circuits and Systems I, vol. 52, pp. 21-31, 2005.
- [25] I.L. Syllaïos, R.B. Staszewski, P.T. Balsara, " *Time-Domain Modeling of an RF All-Digital PLL*", IEEE Transactions on Circuits and Systems II, vol. 55, pp. 601-605, 2008.
- [26] G. Feng, " *Stability analysis of piecewise discrete-time linear systems*", IEEE Transactions on Automatic Control, vol. 47, pp. 1108-1112, 2002.

- [27] M. Johansson, and A. Rantzer, "*Computation of piecewise quadratic Lyapunov functions for hybrid systems*", IEEE Transactions on Automatic Control, vol. 43, pp. 555–559, 1998.
- [28] F.M. Gardner, "*Phaselock Techniques*", Wiley-Interscience, New York, 1979.
- [29] E. Zianbetov et al., "*Design and VHDL modeling of all-digital PLLs*", 8th IEEE International NEWCAS Conference, pp. 293-296, Montreal, 2010.
- [30] B. Mohar, "*Some Applications of Laplace Eigenvalues of Graphs*", in Graph Symmetry: Algebraic Methods and Applications, Kluwer, 1997, pp. 225-275.
- [31] J-M. Akre et al., "*A Design Approach for Networks of Self-Sampled All-Digital Phase-Locked Loops*", 20th European Conference on Circuit Theory and Design (ECCTD), pp. 725-728, 2011.
- [32] F. O'Mahony et al., "*A 10-GHz Global Clock Distribution using Coupled Standing Wave Oscillators*", IEEE Journal of Solid-State Circuits, vol.38, n.11, November 2003.
- [33] J.R.C. Piqueira et al., "*Synchronous State in a Fully Connected Phase-Locked Loop network*", Hindawi Publishing Corporation, Mathematical Problems in Engineering, vol. 2006, pp. 1-12, 2006.
- [34] A.M. Bueno et al., "*Fully Connected PLL Networks: How Filter Determines the Number of Nodes*", Hindawi Publishing Corporation, Mathematical Problems in Engineering, vol. 2009, pp. 1-13, 2009.
- [35] N.L. Margaris, "*Theory of the Non-linear Analog Phase Locked Loop*", LNCIS, Springer, 2004.
- [36] P. Clerc, "*Système dynamique et plan de phase : étude d'une PLL*", Bulletin de l'Union des physiciens ISSN 0366-3876, pp. 111-130, 2004.
- [37] A.M. Harb and B.A. Harb, "*Chaos control of third-order phase-locked loops using backstepping nonlinear controller*", Chaos, Solitons & Fractals, Vol. 20, pp. 719-723, 2004.
- [38] L.H.A. Monteiro et al., "*Route to chaos in a third-order phase-locked loop network*", Signal Processing, Vol. 89, pp. 1678–1682, 2009.
- [39] B.A. Harb and A.M. Harb, "*Chaos and bifurcation in a third-order phase-locked loop*", Chaos, Solitons & Fractals, Vol. 19, pp. 667-672, 2004.
- [40] T. Banerjee and B.C. Sarkar, "*Chaos and bifurcation in a third-order phase-locked loop*", AEU-International journal of Electronics and Communications, Vol. 62, pp. 86-91, 2008.

- [41] D.Y. Abramovitch, "*Analysis and Design of a Third Order Phase-Lock Loop*", IEEE Military Communications Conference, Vol. 2, pp. 455-459, 1988.
- [42] L.H.A. Monteiro et al., "*Bifurcation analysis for third-order phase-locked loops*", IEEE Signal Processing Letters, pp. 494-496, 2004.
- [43] J.R.C. Piqueira, "*Using bifurcations in the determination of lock-in ranges for third-order phase-locked loops*", Communications in Nonlinear Science and Numerical Simulation, Vol. 14, pp. 2328-2335, 2009.
- [44] H.A. Tanaka, et al., "*Synchronization Limit and Chaos Onset in Mutually Coupled Phase-Locked Loops*", IEEE Asia-Pacific Conference on Circuits and Systems, pp. 317-321, 1994.
- [45] T. Endo and L.O. Chua, "*Chaos in mutually coupled phase-locked loops*", IEEE International Symposium on Circuits and Systems, Vol. 1, pp. 399-402, 1990.
- [46] T. Endo and L.O. Chua, "*Chaos from phase-locked loops*", IEEE Transactions on Circuit and Systems, pp. 987-1003, 1988.
- [47] W.C. Lindsay and C.M. Chie, "*A survey of digital phase-locked loops*", Proc. IEEE, Vol. 69, pp. 410-431, 1981.
- [48] W.T. Greer, "*Digital phase-locked loops move into analog territory*", Electronic Design, pp. 95-100, 1982.
- [49] J.C. Lee and C.K. Un, "*Performance analysis of digital tanlock loop*", IEEE Transactions on Communications, Vol. COM-30, pp. 2398-2411, 1982.
- [50] S. Body, L. Ghaoui, E. Feron, and V. Balakrishnan, "*Linear Matrix Inequalities in Systems and Control Theory*". Philadelphia, PA: SIAM, 1994.
- [51] A.A. Ahmadi and P.A. Parrilo, "*Non-monotonic Lyapunov functions for stability of discrete time nonlinear and switched systems*", 47th IEEE Conference on Decision and Control, pp. 614-621, 2008.
- [52] J.L. Stensby, "*Phase-Locked Loops: Theory and Applications*", CRC Press, New York, 1997.
- [53] W.E. Larimore, "*Synthesis of digital phase-locked Loops*", 1968 EASCON Rec., pp. 14-20, 1968.
- [54] W.E. Larimore, "*Design and performance of a second-order digital phase-locked loop*", Proceedings of the Symp. Computer Processing in Communications, pp. 343-357, 1969.

- [55] R.J. Baker, *"CMOS: Circuit Design, Layout, and Simulation"*, Third Edition, IEEE Series on Microelectronic Systems, IEEE Press, 2010.
- [56] G. Bianchi, *"Phase-Locked Loop Synthesizer Simulation"*, McGraw-Hill Electronic Engineering, 2005.
- [57] V.F. Kroupa, *"Phase Lock Loops and Frequency Synthesis"*, John Wiley & Sons, Ltd, 2003.
- [58] D. Abramovitch, *"Lyapunov redesign of classical digital phase-lock loops"*, Proceedings of the 2003 American Control Conference, Vol.3, pp. 2401-2406, 2003.
- [59] F.M. Gardner, *"Charge-Pump Phase-Lock Loops"*, IEEE Transactions on Communications, Vol. COM-28, pp. 1849-1858, 1980.
- [60] P. Acco and al., *"Behavioral Modeling of Charge Pump Phase Locked Loops"*, Proceedings of the 1999 IEEE International Symposium on Circuits and Systems, Vol.1, pp. 375-378, 1999.
- [61] P.M. Levine and G.W. Roberts, *"A high-resolution flash time-to-digital converter and calibration scheme"*, Proceedings. International Test Conference, pp. 1148-1157, 2004.
- [62] E. Zianbetov and al., *"A Digitally Controlled Oscillator in a 65-nm CMOS process for SoC clock generation"*, 2011 IEEE International Symposium on Circuits and Systems, pp. 2845-2848, 2011.
- [63] J.N. Akre, J. Juillard, D. Galayko, and E. Colinet *"Synchronization Analysis of Networks of Self-Sampled All-Digital Phase-Locked Loops"*, IEEE Transactions on Circuits and Systems I, pp. 708-720, 2011.
- [64] D. Abramovitch, *"Phase-Locked Loops: A Control Centric Tutorial"*, Proceedings of the 2002 American Control Conference, 2002.
- [65] D. H. Wolaver, *"Phase-Locked Loop Circuit Design"*, Advanced Reference Series & Biophysics and Bioengineering Series, Englewood Cliffs, New Jersey 07632: Prentice Hall, 1991.
- [66] Z. Gajic and M.T.J. Qureshi, *"Lyapunov Matrix Equation in System Stability and Control"*, Academic Press, Georgia Institute of Technology, Mathematics in Science and Engineering, Vol.195, 1995.
- [67] H. Khalil, *"Nonlinear Systems"*, Macmillan, New York, 1992.

- [68] C. Shan et al., "*FPGA Implementation of Reconfigurable ADPLL Network for Distributed Clock Generation*", International Conference on Field-Programmable Technology (FPT), pp. 1-4, 2011.
- [69] M. Javidan et al., "*A novel technique to reduce the metastability of Bang-Bang Phase Frequency Detectors*", 2011 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 2577-2580, 2011.
- [70] M. Javidan et al., "*All-digital PLL array provides reliable distributed clock for SOCs*", 2011 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 2589-2592, 2011.
- [71] A. Korniienko et al., "*Control law synthesis for distributed multi-agent systems: Application to active clock distribution networks*", American Control Conference (ACC), pp. 4691-4696, 2011.
- [72] A. Korniienko et al., "*A clock network of distributed ADPLLs using an asymmetric comparison strategy*", Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 3212-3215, 2010.
- [73] A. Korniienko et al., " *H_∞ loop shaping control for distributed PLL network*", Ph.D. Research in Microelectronics and Electronics, PRIME 2009, pp. 336-339, 2009.
- [74] N.M. Freris and P.R. Kumar, "*Fundamental Limits on Synchronization of Affine Clocks in Networks*", 46th IEEE Conference on Decision and Control, pp. 921-926, New Orleans, 2007.
- [75] J.R.C. Piqueira et al., "*Linear Approach for Synchronous State Stability in Fully Connected PLL networks*", Hindawi Publishing Corporation, Mathematical Problems in Engineering, vol. 2008, pp. 1-13, 2008.
- [76] G. Goldsztein and S.H. Strogatz, "*Stability of synchronization in Networks of Digital Phase-Locked Loops*", International Journal of Bifurcation and Chaos, Vol. 5, pp. 983-990, 1995.
- [77] A. Teplinsky, R. Flynn and O. Feely, "*Limit Cycles in Bang-Bang Phase-Locked Loops*", International Symposium on Circuits and Systems (ISCAS 2006), pp 4074-4077, 2006.
- [78] A.S. Pikovsky and M.G. Rosenblum, "*Phase Synchronization of Regular and Chaotic Self-Sustained Oscillators*", Theory and Application, eds. A. Pikovsky and Yu, pp. 187-219, 2003.

- [79] O. Aliaei, "*Synchronization and phase synthesis using PLL neural networks*", IEEE International Symposium on Circuits and Systems (ISCAS 2006), pp. 3257-3260, 2006.
- [80] S.H. Strogatz, "*Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering (Studies in Nonlinearity)*", 1st edition, Perseus Books Group, ISBN 0738204536, 2001.
- [81] V. Rasvan and R. Stefan, "*Systèmes non linéaires, théorie et applications*", Germes Science Publications, Lavoisier 2007.
- [82] G.A. Leonov, "*Strange attractors and classical stability theory*", St. Petersburg University Press, 2008.
- [83] G. Osipov, J. Kurths, and C. Zhou, "*Synchronization in Oscillatory Networks*", Springer, Berlin, 2007.
- [84] P. Madoglio et al., "*Quantization Effects in All-Digital Phase-Locked Loops*", IEEE Transactions on Circuits and Systems II: Express Briefs, Vol. 54, No. 12, 2007.
- [85] Hunt B.R. et al., "*A Guide to MATLAB - for Beginners and Experienced Users*", Cambridge University Press, Second edition, 2006.
- [86] M. Kihara, S. Ono and P. Eskelinen, "*Digital Clocks for Synchronization and Communications*", Artech House Telecommunications Library, Norwood, MA 02062, 2003.
- [87] S-M. Kang and Y. Leblebici, "*CMOS Digital Integrated Circuits - Analysis and Design*", WCB McGraw-Hill, 2003.
- [88] P.C. Parks and V. Hahn, "*Stability Theory*", Prentice Hall, 1992.
- [89] S.R. Al-Araji, Z.M. Hussain and M.A. Al-Qutayri, "*Digital Phase Lock Loops - Architectures and Applications*", Springer, 2006.
- [90] S. Engelberg, "*Digital Signal Processing - An Experimental Approach*", Springer, 2008.
- [91] A. Barrat, M. Barthélemy and A. Vespignani, "*Dynamical Processes on Complex Networks*", Cambridge University Press, 2008.
- [92] V.G. Ivancevic and T.T. Ivancevic, "*Geometrical Dynamics of Complex Systems - A Unified Modelling Approach to Physics, Control, Biomechanics, Neurodynamics and Psycho-Socio-Economical Dynamics*", Springer, International Series on Microprocessor-Based And Intelligent Systems Engineering, Vol.31, 2006.
- [93] R.E. Best, "*Phase-Locked Loops - Design, Simulation, and Applications*", McGraw-Hill, Fourth Edition, 1999.

- [94] D.R. Stephens, *"Phase-Locked Loops for Wireless Communications - Digital, Analog and Optical Implementations"*, Second Edition, Kluwer Academic Publishers, 2002.
- [95] K. Shu and E. Sanchez-Sinencio, *"CMOS PLL Synthesizers: Analysis and Design"*, Springer, 2005.
- [96] J. Reichardt, *"Structure in Complex Networks"*, Lecture Notes in Physics 766, Springer, 2009.
- [97] A. Pikovsky, M. Rosenblum and J. Kurths, *"Synchronization - A Universal Concept in Nonlinear Sciences"*, Cambridge Nonlinear Science Series 12, Cambridge University Press, 2001.
- [98] B. A. Shenoi, *"Introduction to Digital Signal Processing and Filter Design"*, Wiley Interscience, 2006.
- [99] C.W. Wu, *"Synchronization in Complex Networks of Nonlinear Dynamical Systems"*, World Scientific, 2007.
- [100] S.T. Karris, *"Digital Circuit Analysis and Design with Simulink® Modeling and Introduction to CPLDs and FPGAs"*, Second Edition, Orchard Publications.
- [101] S. Mendel, C. Vogel and N.D Dalt, *"A Phase-Domain All-Digital Phase-Locked Loop Architecture without Reference Clock Retiming"*, IEEE Transactions on Circuits and Systems II: Express Briefs, Vol.56, Issue 11, pp. 860-864, Nov 2009.
- [102] S. Tertinek, J.P. Gleeson, and O. Feely, *"Statistical Analysis of First-Order Bang-Bang Phase-Locked Loops Using Sign-Dependent Random Walk Theory"*, IEEE Transactions on Circuits and Systems I, Vol. 57, pp. 2367-2380, 2010.
- [103] B. Chun and M.P. Kennedy, *"Statistical Properties of First-Order Bang-Bang PLL With Nonzero Loop Delay"*, IEEE Transactions on Circuits and Systems II: Express Briefs, Vol.55, pp. 1016-1020, 2008.
- [104] S.H. Strogatz, *"Spontaneous Synchronization in Nature"*, IEEE International Frequency Control Symposium, pp. 2-4, 1997.
- [105] M.G. Earl and S.H. Strogatz, *"Synchronization in oscillator networks with delayed coupling: A stability criterion"*, Physical Review, pp. 036204-1-036204-4, 2003.
- [106] X. Wang et al., *"Simulink Model for All-Digital-Phase-Locked-Loop"*, IEEE International Workshop on Radio-Frequency Integration Technology, pp. 70-73, 2007.
- [107] H.Sutoh, K. Yamakoshi, and M. Ino, *"Circuit Technique for Skew-Free Clock Distribution"*, IEEE Custom Integrated Circuits Conference, pp. 8.7.1-8.7.4, 1995.

- [108] G-C. Hsieh and J.C. Hung, "*Phase-Locked Loop Techniques – A Survey*", IEEE Transactions on Industrial Electronics, Vol. 43, pp. 609-615, 1996.
- [109] J. Garodnick, J. Greco, and D.L. Schilling, "*Response of an All Digital Phase-Locked Loop*", IEEE Transactions on Communications, pp. 751-764, 1974.

- [T1] F.P. O'Mahony "*10 GHz Global Clock Distribution Using Coupled Standing-Wave Oscillators*", PhD Thesis, Stanford University, 145p, 2003.
- [T2] F.D. Hutu, "*Étude de la synchronisation d'oscillateurs : application aux antennes actives*", Thèse de doctorat, Université de Poitiers, 177p, 2007.
- [T3] C. WARE, "*Récupération d'horloge par boucle à verrouillage de phase utilisant le mélange à quatre ondes dans un amplificateur optique à semi-conducteurs*", Thèse de doctorat, École Nationale Supérieure des Télécommunications, 135p, 2003.
- [T4] P. Acco, "*Étude de la boucle à verrouillage de phase par impulsions de charge - Prise en compte des aspects hybrides*", Thèse de doctorat, Institut National des Sciences Appliquées de Toulouse, 142p, 2003.
- [T5] A. Kornienko, "*Réseau de PLLs distribuées pour synthèse automatique d'horloge de MPSOCs synchrones*", Thèse de doctorat, CEA, LETI, MINATEC, 252p, 2011.

LISTE DES FIGURES ET TABLEAUX

Fig. 0.1 – réseau de distribution d’horloge de 16 nœuds, réalisé par Gutnik et Chandrakasan.....	4
Fig. 1.1 – Schéma classique d’une PLL analogique.....	10
Fig. 1.2 – Filtre de boucle.....	12
Fig. 1.3. – Plan de phase de la solution de l’équation (1.22).....	16
Fig. 1.4. – Plan de phase de la solution de l’équation (1.22).....	18
Fig. 1.5. – Plan de phase de la solution de l’équation (1.22) pour $P_3 = 0.8$	18
Fig. 2.1 – Schéma de base de l’ADPLL.....	22
Fig. 2.2 – ADPLL avec un comparateur XOR classique.....	23
Fig. 2.3 – Détection de l’erreur de phase avec une porte logique XOR.....	24
Fig. 2.4 – Caractéristique du détecteur XOR.....	24
Fig. 2.5 – Fréquence de l’horloge locale en fonction du code de contrôle y_i	25
Fig. 2.6 –Schéma simplifié de l’ADPLL XOR de Greer.....	26
Fig. 2.7 – Schéma d’un comparateur de phase et de fréquence à bascules D.....	27
Fig. 2.8 – Fonctionnement en comparateur de phase du détecteur à bascules : (a) Horloge de référence en avance par rapport à l’horloge locale, (b) Horloge de référence en retard par rapport à l’horloge locale.....	28
Fig. 2.9 – Fonctionnement en comparateur de fréquence du détecteur à bascules : (a) Fréquence de l’horloge de référence supérieure à celle de l’horloge locale, (b) Fréquence de l’horloge de référence inférieure à celle de l’horloge locale.....	28
Fig. 2.10 – Caractéristique de détecteur de phase et de fréquence à bascules D.....	29
Fig. 2.11 – Modèle d’ADPLL avec un DPD à bascules D.....	29
Fig. 2.12 – Modèle Simulink d’une ADPLL avec un DPD à bascules D.....	30
Fig. 2.13 – Simulation transitoire d’une ADPLL avec un DPD à bascules D pour un couple de coefficients $A = 0.2282$ et $B = -0.2244$. En vert : Impulsions de durée proportionnelle à l’erreur de phase en sortie du DPD, En rouge : Signal d’erreur temporelle en sortie du TDC. En bleu : Signal de commande du DCO.....	31
Fig. 2.14 – Schéma de base de la SS-ADPLL.....	31
Fig. 2.15 – Automate à 4 états du détecteur de phase et de fréquence bang-bang.....	33
Fig. 2.16 – Schéma du détecteur de phase et de fréquence.....	33
Fig. 2.17 – Une implémentation physique possible d’un TDC.....	34
Fig. 2.18 – Caractéristiques d’un TDC : (a) linéaire, (b) non linéaire.....	35
Fig. 2.19 – Caractéristique du DPD HODISS.....	35
Fig. 2.20 – Schéma d’un filtre PI typique.....	36
Fig. 2.21 – Diagramme de Bode d’un correcteur PI pour un couple de coefficient donné.....	37
Fig. 2.22 – Implémentation du DCO d’une ADPLL.....	38
Fig. 2.23 – Évolution de la fréquence du DCO en fonction du code digital de contrôle.....	39
Fig. 2.24 – Bruit de phase pour une fréquence centrale de 2GHz.....	39
Fig. 2.25 – Modèle d’une ADPLL sans auto-échantillonnage.....	40
Fig. 2.26 – Domaine de stabilité d’une ADPLL classique sans auto-échantillonnage.....	42
Fig. 2.27 – Zone isochrone avec en son centre son horloge locale (à gauche) et ensemble de zones isochrones échangeant entre elles des informations d’erreurs.....	43
Fig. 2.28 – Diagramme bloc d’une SS-ADPLL passéiste.....	44
Fig. 2.29 – Chronogramme des signaux en présence de l’entrée de la SS-ADPLL jusqu’à l’entrée du filtre.....	45
Fig. 2.30 – Mise en œuvre du filtre PI de type A.....	47
Fig. 2.31 - Domaine de stabilité d’une SS-ADPLL passéiste de type A. La zone grisée correspond aux valeurs de coefficients du filtre pour lesquelles les simulations ont montré que le SS-ADPLL synchronise.....	51
Fig. 2.32 – Mise en œuvre du filtre PI de type B.....	51
Fig. 2.33a. Domaine de stabilité d’une SS-ADPLL passéiste de type A. La zone grise claire a été obtenue en calculant les pôles du système moyen, la zone grise foncée à celle obtenue par simulations transitoires et la zone noire est l’intersection des deux. Les lignes en pointillé représentent les limites du domaine de stabilité d’une ADPLL idéale gouvernée par l’équation (2.17).....	53
Fig. 2.33.b. Domaine de stabilité d’une SS-ADPLL passéiste de type B. La zone grise claire a été obtenue en calculant les pôles du système moyen, la zone grise foncée à celle obtenue par simulations transitoires et la zone noire est l’intersection des deux. Les lignes en pointillé représentent les limites du domaine de stabilité d’une ADPLL idéale gouvernée par l’équation (2.17).....	54

Fig. 3.1 - Diagramme de transition d'une SS-ADPLL passéiste de type A (a) et celui d'une SS-ADPLL passéiste de type B (b).....	64
Fig. 3.2 - Domaine de stabilité d'une SS-ADPLL passéiste de type B. La zone grisée est obtenue par simulation transitoire et la zone noire est obtenue par recherche de PQLFs en résolvant des LMIs.	65
Fig.3.3 - Bassins d'attraction d'une SS-ADPLL passéiste de type A. La zone blanche (resp. noire) correspond aux conditions initiales conduisant à des trajectoires stables (resp. instables).	66
Fig. 4.1 - Réseau autonome de 2 SS-ADPLLs.	71
Fig. 4.2 - Domaine de stabilité (en grisé) d'un réseau de 2 SS-ADPLLs passéistes de type A.	74
Fig. 4.3 - Domaine de stabilité (en grisé) d'un réseau de 2 SS-ADPLLs passéistes de type B.	74
Fig. 4.4 - Domaine de stabilité (en grisé) d'un réseau de 2 SS-ADPLLs nihilistes de type A.	75
Fig. 4.5 - Domaine de stabilité (en grisé) d'un réseau de 2 SS-ADPLLs nihilistes de type B.	75
Fig. 4.6 - Deux réseaux cartésiens de 4 et 9 SS-ADPLLs. A chaque nœud, l'erreur totale est définie comme étant la valeur moyenne des sorties des DPDs.	77
Fig. 4.6a - Propagation des erreurs dans chacune des branches dans un réseau cartésien autonome de 9 nœuds.	79
Fig. 4.7. Un réseau cartésien incomplet de 7 SS-ADPLLs avec le vecteur v^T correspondant.	84
Fig. 4.8 - Domaine de stabilité d'un réseau cartésien complet de 9 SS-ADPLLs passéistes de type A. Les lignes en pointillés représentent les limites du domaine de stabilité dérivé de l'équation maitresse (4.47a).	89
Fig. 4.9 - Domaine de stabilité d'un réseau cartésien complet de 16 SS-ADPLLs passéistes de type A. Les lignes en pointillés représentent les limites du domaine de stabilité dérivé de l'équation maitresse (4.47a).	89
Fig. 4.10 - Domaine de stabilité d'un réseau cartésien complet de 256 SS-ADPLLs passéistes de type A. Les lignes en pointillés représentent les limites du domaine de stabilité dérivé de l'équation maitresse (4.47a).	89
Fig. 4.11 - La réponse d'un réseau de 4 SS-ADPLLs passéistes de type A (lignes noires en gras) comparé à celle de son réseau moyen (lignes colorées en pointillés). Simulation pour $K_1=0.8$ et $K_2=-0.7$	90
Fig. 4.12. La réponse d'un réseau de 4 SS-ADPLLs passéistes de type B (lignes noires en gras) est très similaire de celle de son réseau moyen (en couleurs). Simulation pour $K_1=0.8$ et $K_2=-0.7$	90
Fig. 4.13 - Domaine de stabilité (zone grise) d'un réseau de 9 SS-ADPLLs nihilistes de type A. Les lignes en pointillés représentent les limites du domaine de stabilité dérivé de l'équation maitresse (4.54a).	92
Fig. 4.14 - Domaine de stabilité (zone grise) d'un réseau de 16 SS-ADPLLs nihilistes de type A. Les lignes en pointillés représentent les limites du domaine de stabilité dérivé de l'équation maitresse (4.54a).	93
Fig. 4.15 - Domaine de stabilité (zone grise) d'un réseau de 9 SS-ADPLLs nihilistes de type B. Les lignes en pointillés représentent les limites du domaine de stabilité dérivé de l'équation maitresse (4.54b).	93
Fig. 4.16 - Domaine de stabilité (zone grise) d'un réseau de 16 SS-ADPLLs nihilistes de type B. Les lignes en pointillés représentent les limites du domaine de stabilité dérivé de l'équation maitresse (4.54b).	93
Fig. 4.17 - Domaine de stabilité (zone grise) d'un réseau de 64 SS-ADPLLs nihilistes de type B. Les lignes en pointillés représentent les limites du domaine de stabilité dérivé de l'équation maitresse (4.54b).	94
Fig. 4.18 - Position centrale de l'équation maitresse permettant d'approximer un réseau de SS-ADPLLs par son système moyen linéaire.	95
Fig. 4.19 - Domaines de stabilité du système moyen correspondant à un réseau de 2 SS-ADPLLs passéistes de type B (a) et un réseau de 4 SS-ADPLLs passéistes de type B (b) en fonction de la marge de module.	95
Fig. 4.20 - Domaines de stabilité du système moyen correspondant à un réseau de 9 SS-ADPLLs passéistes de type B (a) et un réseau de 16 SS-ADPLLs passéistes de type B (b) en fonction de la marge de module.	96
Fig. 4.21 - Erreur totale pour un nœud arbitraire d'un réseau moyen de 16 SS-ADPLLs passéistes de type B pour un couple de coefficients optimums (a) et non optimum (b).	96
Fig. 4.22 - Modélisation de la prise en compte des erreurs de quantification dans un nœud « moyenné » de SS-ADPLL passéiste.	98
Fig. 4.23 - Module de la fonction de transfert $H_b(z)$ pour un réseau de 16 SS-ADPLLs passéistes de type B.	99
Fig. 4.24 - Réseau cartésien de 9 SS-ADPLLs avec une référence.	100
Fig. 5.1 - Domaine de stabilité d'un réseau de 4 SS-ADPLLs passéistes de type B.	104
Fig. 5.2 - Domaine de stabilité d'un réseau de 9 SS-ADPLLs passéistes de type B.	104
Fig. 5.3 - Domaine de stabilité d'un réseau de 81 SS-ADPLLs passéistes de type B.	104
Fig. 5.4 - Domaine de stabilité d'un réseau de 256 SS-ADPLLs passéistes de type B.	104
Fig. 5.5 - Domaine de stabilité d'un réseau de 16 SS-ADPLLs passéistes de type B.	105
Fig. 5.6 - Domaine de stabilité d'un réseau de 16 SS-ADPLLs passéistes de type B avec $Q=0.001$	105
Fig. 5.7 - Domaine de stabilité d'un réseau de 16 SS-ADPLLs passéistes de type B avec $Q=0.004$	105
Fig. 5.8 - Domaine de stabilité d'un réseau de 16 SS-ADPLLs passéistes de type B avec $Q=0.012$	106
Fig. 5.9 - Domaine de stabilité d'un réseau de 16 SS-ADPLLs passéistes de type B avec $Q=0.015$	106
Fig. 5.10 - Domaine de stabilité d'un réseau de 16 SS-ADPLLs passéistes de type B avec une référence.	107
Fig. 5.11 - Derniers fronts (50) des horloges en sorties d'un réseau de 16 SS-ADPLLs de type B.	108
Fig. 5.12 - Erreurs totales en sortie d'un réseau de 16 SS-ADPLLs de type B avec référence.	108

Tableau 5.1 – Tableau récapitulatif des paramètres ASIC, FPGA et des paramètres normalisés.	109
Fig. 5.13 – Signaux de sortie d'un réseau de 16 SS-ADPLLs, on n'observe pas de modelock (configuration : bidirectionnelle, fréquences centrales identiques).	111
Fig. 5.14 – Signaux de sortie d'un réseau de 16 SS-ADPLLs, phase initiale de la configuration dynamique (mode unidirectionnel), fréquences centrales différentes, pas de modelock.	112
Fig. 5.15 – Signaux de sortie d'un réseau de 16 SS-ADPLLs, phase secondaire de la configuration dynamique (mode bidirectionnel), fréquences centrales différentes, pas de modelock et erreurs plus petites.	113
Fig. A.1.1 - Réseau complètement connecté de 5 nœuds.	122
Fig. A.1.2 - Domaine de stabilité (en grisé) d'un réseau complètement connecté pour $N = 2$	122
Fig. A.1.3 - Domaine de stabilité (en grisé) d'un réseau complètement connecté pour $N = 4$	123
Fig. A.1.4 - Domaine de stabilité (en grisé) d'un réseau complètement connecté pour $N = 16$	123
Fig. A.1.5 - Réseau cartésien de 9 nœuds.	124
Fig. A.1.6 - Domaine de stabilité (en grisé) d'un réseau quadratique de taille quelconque.	124
Fig. A.1.7 - Répartition des valeurs propres de $\mathbf{\Gamma}$ à l'intérieur du cercle unité pour un réseau quadratique de taille $N = 9$ (croix) ou $N = 36$ (ronds).	125
Fig. A.1.8 - Répartition des valeurs propres de $\mathbf{\Gamma}$ à l'intérieur du cercle unité pour un réseau complètement connecté de taille $N = 9$ (croix) ou $N = 36$ (ronds). Pour un réseau de taille beaucoup plus importante, les valeurs propres non-triviales de $\mathbf{\Gamma}$ sont très proches des ronds.	125
Fig. A.2.1 Domaine de stabilité d'un réseau de 4 SS-ADPLLs passésistes de type A	126
Fig. A.2.2 Domaine de stabilité d'un réseau de 9 SS-ADPLLs passésistes de type A	126
Fig. A.2.3 Domaine de stabilité d'un réseau de 81 SS-ADPLLs passésistes de type A	126
Fig. A.2.4 Domaine de stabilité d'un réseau de 256 SS-ADPLLs passésistes de type A	126
Fig. A.2.5 Domaine de stabilité d'un réseau de 16 SS-ADPLLs passésistes de type A	127

RÉSUMÉ

Le projet HODISS dans le cadre duquel s'effectue nos travaux adresse la problématique de la synchronisation globale des systèmes complexes sur puce (System-on-Chip ou SOC, par exemple un multiprocesseur monolithique). Les approches classiques de distribution d'horloges étant devenues de plus en plus obsolètes à cause de l'augmentation de la fréquence d'horloge, l'accroissement des temps de propagation, l'accroissement de la complexité des circuits et les incertitudes de fabrication, les concepteurs s'intéressent (pour contourner ces difficultés) à d'autres techniques basées entre autres sur les oscillateurs distribués. La difficulté majeure de cette dernière approche réside dans la capacité d'assurer le synchronisme global du système. Nous proposons un système d'horlogerie distribuée basé sur un réseau d'oscillateurs couplés en phase. Pour synchroniser ces oscillateurs, chacun d'eux est en fait une boucle à verrouillage de phase qui permet ainsi d'assurer un couplage en phase avec les oscillateurs des zones voisines. Nous analysons la stabilité de l'état synchrone dans des réseaux cartésiens identiques de boucles à verrouillage de phase entièrement numériques (ADPLLs). Sous certaines conditions, on montre que l'ensemble du réseau peut synchroniser à la fois en phase et en fréquence. Un aspect majeur de cette étude réside dans le fait que, en l'absence d'une horloge de référence absolue, le filtre de boucle dans chaque ADPLL est piloté par les fronts montants irréguliers de l'oscillateur local et, par conséquent, n'est pas régi par les mêmes équations d'état selon que l'horloge locale est avancée ou retardée par rapport au signal considéré comme référence. Sous des hypothèses simples, ces réseaux d'ADPLLs dits "auto-échantillonnés" peuvent être décrits comme des systèmes linéaires par morceaux dont la stabilité est notoirement difficile à établir. L'une des principales contributions que nous présentons est la définition de règles de conception simples qui doivent être satisfaites sur les coefficients de chaque filtre de boucle afin d'obtenir une synchronisation dans un réseau cartésien de taille quelconque. Les simulations transitoires indiquent que cette condition nécessaire de synchronisation peut également être suffisante pour une classe particulière d'ADPLLs "auto-échantillonnés".

ABSTRACT

The HODISS project, context in which this work is achieved, addresses the problem of global synchronization of complex systems-on-chip (SOCs, such as a monolithic multiprocessor). Since the traditional approaches of clock distribution are less used due to the increased clock frequency, increased delay, increased circuit complexity and uncertainties of manufacture, designers are interested (to circumvent these difficulties) to other techniques based among others on distributed synchronous clocks. The main difficulty of this latter approach is the ability to ensure the overall system synchronization. We propose a clock distribution system based on a network of phase-coupled oscillators. To synchronize these oscillators, each is in fact a phase-locked loop which allows to ensure a phase coupling with the nearest neighboring oscillators. We analyze the stability of the synchronized state in Cartesian networks of identical all-digital phase-locked loops (ADPLLs). Under certain conditions, we show that the entire network may synchronize both in phase and frequency. A key aspect of this study lies in the fact that, in the absence of an absolute reference clock, the loop-filter in each ADPLL is operated on the irregular rising edges of the local oscillator and consequently, does not use the same operands depending on whether the local clock is leading or lagging with respect to the signal considered as reference. Under simple assumptions, these networks of so-called "self-sampled" all-digital phase-locked-loops (SS-ADPLLs) can be described as piecewise-linear systems, the stability of which is notoriously difficult to establish. One of the main contributions presented here is the definition of simple design rules that must be satisfied by the coefficients of each loop-filter in order to achieve synchronization in a Cartesian network of arbitrary size. Transient simulations indicate that this necessary synchronization condition may also be sufficient for a specific class of SS-ADPLLs.